Data Modeling in the Sciences

Applications, Basics, Computations

Steve Pressé and Ioannis Sgouralis

(This draft was last modified on February 16, 2021)

Preface

Data analysis courses that go beyond teaching elementary topics, such as fitting residuals, are rarely offered to students in the Natural Sciences. As a result, data analysis, much like programming, is something often learned and improvised "on the job". Yet, with an explosion of experimental methods generating large quantities of diverse data, we believe that students and researchers alike would benefit from a clear presentation of methods of analysis many of which have only become feasible due to the practical needs and computational advances of the last decade or two.

The framework for data analysis that we provide here is inspired by exciting new developments in Data Science, Machine Learning and Statistics in a language accessible to the broader community of Natural scientists. As such, this text is ambitiously aimed at making topics such as statistical inference, computational modeling and simulation approachable to the Natural Sciences.

It is also a goal of ours, if nothing else, to help develop an appreciation for data-driven modeling and what data analysis choices are available to us alongside what approximations are inherent to the choices explicitly or implicitly made. We do so because theory in the Physical Sciences has traditionally provided limited emphasis on data-driven approaches. Indeed, the prevailing philosophy is that models are first proposed and then verified or otherwise disproven by experiments. But this approach is not data-centric. Nor is it rigorous except for the cleanest of experimental data sets as one's perceived choice in how to compare models and experiments may have dramatic consequences in whether the model is ultimately falsified. As we move toward monitoring events on smaller or faster timescales or complex events otherwise sparsely sampled, examples of clean data are already few and far between.

We designed the text as a self-contained single semester course in data analysis, statistical modeling and inference. We have now use it in teaching three times at Arizona State University (2017-) to first year Chemistry and Physics graduate students as well as upper-level undergraduates. While the text is appropriate for upper-level undergraduates in the Physical Sciences, its intended audience is at the master's level. The concepts presented herein are self-contained though a basic course in computer programming and prior knowledge of undergraduate level calculus is assumed.

Our text places equal emphasis on explaining the foundations of existing methods and their implementation. It correspondingly places little emphasis on formal proofs and research active topics at the forefront of data analysis yet to be settled. Along core sections, we have interspersed sections and topics designated by an asterisk. These contain more advanced materials that may be included at the instructor's discretion and are otherwise not necessary upon a first reading.

The text begins with a survey of modeling concepts to motivate the problem of parameter estimation from given data. This leads to a discussion of frequentist and Bayesian inference tools. Along the way, we introduce computational techniques including Monte Carlo methods that are necessary for a comprehensive exposition of the most recent advances. The second half of the text is devoted to specific models starting from basic mixture models followed by Gaussian processes, the hidden Markov model, its adaptations as well as its generalization to state-space models and continuous time representations.

Finally, we made clear choices on what topics to include in the book. These were sometimes based on personal interest though, most often, these choices were based on what we believe is most relevant. To keep our presentation streamlined, however, we have excluded many topics. Some of these are topics that we perceive as easier for students to understand after reading this book, such as special cases or otherwise specialized generalizations of the topics covered herein.

Tempe, AZ
Knoxville, TN
February 16, 2021

Thanks

Many thanks to Weiqing Xu for generating figures in the earlier chapters as well as Sina Jazani and Zeliha Kilic for helping write portions of the text. Special thanks to Julian Lee and Corey Weistuch for reading over the text. We also thank the many other members of the Pressé lab and students at Arizona State University taking CHM/PHY 598 ("Unraveling the noise") who have suggested many revisions and identified typos across earlier drafts of the text. Any remaining typos and omissions are ours alone.

Short contents

Contents

*This is an advanced topic and could be skipped on a first reading.

*This is an advanced topic and could be skipped on a first reading.

*This is an advanced topic and could be skipped on a first reading.

†This is an advanced topic and could be skipped on a first reading.

Part I

Concepts from modeling, inference, and computing

Chapter 1

Introduction to probabilistic modeling and inference

By the end of this chapter, we will have presented
- *Data oriented modeling*
- *Random variables and their properties*
- *An overview of inverse problem solving*

1.1 Modeling with data

If experimental observations or, put differently, binaries on a screen were all we ever cared about, then no experiment would require modeling or interpretation and the remainder of this book would be unnecessary. But binaries on a screen do not constitute knowledge. They constitute *data*. Put differently, Quantum Mechanics (or any scientific knowledge) is not self-evident from the pixelated outcome on a camera chip of a modern incarnation of a Young's two-slit interference experiment.

In the Natural Sciences, *models* of physical systems provide mathematical frameworks in which we unify disparate pieces of information. These include conceptual notions such as symmetries, fundamental constituents and other postulates as well as scientific measurements and, even more generally, empirical observations of any form. If we think of direct observations as data in particular, at least for now, we can think of mathematical models as a way of compressing or summarizing data.

These data summaries may be used to make predictions about physical conditions we may encounter in the future, such as in new experiments, or to interpret and describe an underlying physical system already probed in past experiments. For example, with time-ordered data we may be interested in learning equations of motion or kinetic schemes. Or, already knowing a kinetic scheme sufficiently well from past experiments or fundamental postulates, we may only be interested in learning the noise properties of a new piece of equipment on which future experiments will be run. Thus, models may be aimed at discovering new Science as well as at devising careful controls to get a better handle on error bars and, more broadly, even at designing new experiments altogether.

1.1.1 Why do we obtain models from raw data?

Unfortunately, experimental data rarely provide direct insight on the physical conditions and systems of interest. At the very least, measurements are *corrupted* by unavoidable noise and, as a result, models obtained from data are unavoidably probabilistic. So, we ask: *how should we, the scientific community, go about obtaining models from imperfect data?*

Note 1.1: Obtaining models from data

Data can be time and labor intensive to acquire. Perhaps more importantly, every datum in a dataset encodes information. In light of this, we re-pitch our question and ask: *how should we go about obtaining models* efficiently *and* without compromising *the information encoded in the data?*

The key is to start from the data acquired in the experiments and arrive at models with a minimal amount of pre-processing, if at all. This is because obtaining a model from quantities derived from the data, as opposed to directly from the data, is necessarily *equal to* or *worse than* obtaining the model from the data directly since derived quantities contain as much as or less information than the data themselves. For instance, fitting histogrammed data is an information-inefficient and unreliable approach to obtaining models as it demands downsampling via binning and a more or less arbitrary choice of bin sizes.

Besides information efficiency, obtaining models from unprocessed data also has another critical advantage that gets to the heart of scientific practice. While error bars around individual data points may be imperfectly known, they are, by construction, *better characterized* than error bars around derived quantities. Thus error bars around models determined from derived quantities are necessarily only as good as, but often less reliable, than error bars around models determined from the data. Unfortunately, as error bars around derived quantities can become too difficult to compute in practice, they are often ignored altogether. Nevertheless, error bars are a cornerstone of modern scientific research. They not only help quantify reproducibility but they also directly inform error bars around the models obtained and, as such, inspire the formulation of new competing models.

Putting it all together, it becomes clear that a model is *best informed*, and has the *most reliable error bars*, when learned from the data available in as raw a form as accessible from the experiments. This is true so long as it is computationally feasible to obtain models from such raw data and, as we will see in subsequent chapters, we are far from reaching computational bottlenecks in most problems of interest across the Natural Sciences.

1.1.2 Why do we formulate models with random variables?

If there is no uncertainty involved, a physical system is adequately described using deterministic variables. For example, Newtonian mechanics are expressed in terms of momenta, positions, and forces. However, when a system involves any degree of uncertainty, either due to noise, poor characterization of some or all of its constituents, features as of yet unresolved, or otherwise fundamentally stochastic then it is better described by *random variables*. This is true of the probabilistic nature of Quantum Mechanics as well as Statistical Physics and, as we illustrate herewith, also of Data Analysis.

In this book we focus on the latter case; namely *stochastic systems*. Stochasticity in our systems arise due to inherent randomness in the physical phenomena of interest or due to measurement noise or both. We represent observations generated by stochastic systems as *random variables*. This is because, as we will see, random variables are mathematical notions that can reproduce naturally stochastic relationships between uncertain physical phenomena and observations; while, their deterministic counterparts cannot.

Note 1.2: Measurement noise

It is sometimes thought that models with probabilistic formulations are only required when the quantities of interest are inherently probabilistic. Nevertheless, measurement noise corrupts experimental observations irrespective of the quantities themselves being probabilistic or not. Consequently, probabilistic models are *always required* whenever models are informed by experimental output.

Random variables are abstract notions that most often represent numbers or collections of numbers. However, random variables are generic notions and they may also be non-numeric such as: labels for grouping data, *e.g.* group A, group B; words in a text prompt like Google's search engine; functions, *e.g.* trajectories or energy potentials. In all cases, numeric or not, random variables may be *discrete*, *e.g.* dice rolls, coin flips, photon counts or *continuous*, such as temperatures, pressures or distances. Further, random variables may be finite collections of individual quantities, *e.g.* measurements acquired during an experiment or even infinite ones, *e.g.* successive positions on a *Brownian particle's* trajectory. At any rate, random variables have unique properties, which we will shortly explore, that allow us to use them in the construction and evaluation of meaningful probabilistic models.

Commonly, we imagine a random variable, which we denote with W, as being instantiated or assigned a specific value realized at w as a result of performing a measurement which amounts to a *stochastic event*. That is, we think of a measurement output w as a *stochastic realization* of W. In other words, stochastic events encompass randomness inherited through W and influencing the assigned values w.

Stochastic events may encompass *physical* events, like the occurrence of chemical reactions or events in a cell's life cycle. Stochastic events may also encompass *conceptual* events, like an idealized version of a real-life system expressed in terms of fair coin tosses or, even, like instantaneously learning the spin orientation of a faraway particle given a local measurement of another spin to which the first is entangled.

Example 1.1: The photo-electric effect

When a photon falls into certain materials, a photo-electron is sometimes emitted and sometimes not. Such a phenomenon provides the basis for a stochastic event.

In the photo-electric setting, it is often convenient to formulate a random variable W that counts the number of photo-electrons emitted. This random variable may take values $w = 0, 1, 2, \cdots$.

Throughout this introductory chapter, we will distinguish between a random variable W and its realizations, w, *i.e.*, the particular values that W attains or may attain. Due to clarity, we will be strict in our notation but, as we move forward, in the subsequent chapters we will gradually relax our convention since what is meant by w, serving as a proxy for either the value w or the variable W, will be clear from the context of the discussion.

Concretely, to develop a model, we imagine a *prototype experiment* as a sequence of stochastic events that produce N numeric measurements or, more generally, observations of any kind. We typically use w_n to denote the n^{th} observation and use $n = 1, \cdots, N$ to index them.

Example 1.2: Observations in a prototype experiment

Individual observations in our experiment may be scalar values, for example $w_n = 20.1°\text{C}$ or $w_n = 0.74\ \mu\text{m}^3$ for typical measurements of room temperature or an *E. coli*'s volume, respectively.

Individual observations may also be non-numeric, such as $w_n =$ p.R83SfsX15 for descriptions of gene variations.

In general, we do not require that each observation be of the same type; that is, w_1 may be a temperature while w_2 may be a volume.

As we will often do, we gather every observation conveniently together in a list

$$w_{1:N} = \{w_1, w_2, \cdots w_N\}$$

and use subscripts $1 : N$ in $w_{1:N}$ to indicate that we gather every single w_n with an index n between 1 and N. Unless explicitly needed to help draw attention to the subscript, for clarity, we may sometimes suppress this subscript and write simply w for the entire list.

As we have already mentioned, the observations $w_{1:N}$ are better understood as realizations of appropriate random variables $W_{1:N} = \{W_1, W_2, \cdots, W_N\}$. In principle, each observation w_n may be obtained under different conditions, and so $W_{1:N}$ may gather random variables W_n with different properties.

Example 1.3: Photo-electric assessments

Consider a simple experiment where a photon source successively sends N bursts of photons that impinge upon a photo-electric material. Suppose further that, each time a photon burst is sent out from the source, we assess how many photo-electrons are produced. In this case, our assessments can be modeled by an array $W_{1:N} = \{W_1, W_2, \ldots, W_N\}$, where each variable W_n may take values $w_n = 0, 1, 2, \cdots$.

Provided that the intensity of the photon bursts remains constant, it is reasonable to consider all variables in $W_{1:N}$ as maintaining the same properties; however, if the intensity of each burst changes over time, then our model needs to account for different properties for each W_n.

1.1.3 Why do our models have parameters?

Models are themselves mathematical structures and their associated parameters, often highly specialized to particular systems, experiments or experimental setups. Often, the most main objective in Data Analysis becomes

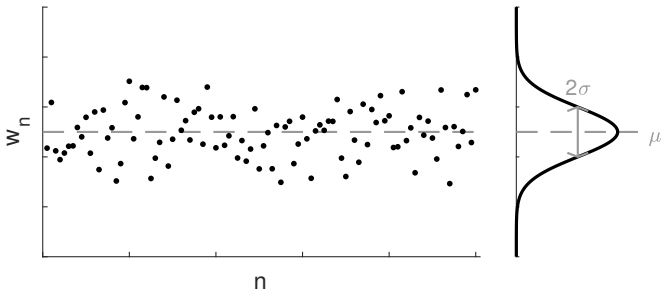


Figure 1.1: On the left hand side we show the output of an experiment after successive trials, n. On the right hand side we find a histogram of the data with very fine bin sizes that assumes the shape of a Gaussian distribution. We denote the mean of this distribution by μ and standard deviation by σ.

the calibration of the model parameters assuming a model structure and provided observed values $w_{1:N}$.

Example 1.4: Normal random variables

The mean of a sequence of identical random variables $W_{1:N}$ is only probabilistically related to each measured value w_n. For the simple example of a normally distributed sequence, what we call the "model" is the normal distribution (often also termed the Gaussian distribution) and its associated parameters; the mean μ and variance σ^2 which indicate the center and spread of the values in $w_{1:N}$, respectively. These are collectively described by the list of model parameters $\theta = \{\mu, \sigma^2\}$. As illustrated in fig. 1.1, and as we will see in detail in later chapters, θ can be estimated from $w_{1:N}$.

In the previous example, the Gaussian forms a model that contains two parameters, namely the mean μ and the variance σ^2, that we gather in θ. More general models may contain K individual parameters and we will often gather them similarly in a list $\theta_{1:K} = \{\theta_1, \theta_2, \ldots, \theta_K\}$.

In a model, the parameters in $\theta_{1:K}$ represent those quantities we care to *estimate,* for example μ and σ^2, and our model is specified when *specific values* are assigned to $\theta_{1:K}$. Thus, obtaining a model, essentially, is the same task as assigning values to $\theta_{1:K}$. Similarly, deriving error bars around the assigned values of $\theta_{1:K}$ is equivalent to deriving error bars around the model.

As we invariably always face some degree of measurement noise, we formulate an experiment's results $w_{1:N}$ as probabilistically related to the parameters $\theta_{1:K}$. In the context of our *prototype experiment*, we incorporate such relations through the random variables $W_{1:N}$ and in the next section we lay down some necessary concepts.

1.2 Working with random variables

Having motivated why we have to learn models from data in the rawest form in which they are acquired, and before we embark on specific learning strategies, we begin by exploring some important notions needed when working with random variables. As we will soon start using random variables to not only to represent measurements W, but also other relevant quantities of our model, we will begin using R to label generic random variables.

1.2.1 How to assign probability distributions

In any model, a random variable R is *drawn* or *sampled from* some *probability distribution*. We label such a distribution with $\mathbb{P}$ and, adopting the language of Statistics, we write

$$R \sim \mathbb{P}.$$

This reads "the random variable R is sampled from the probability distribution $\mathbb{P}$" or "R follows the statistics of $\mathbb{P}$".

In a statistical formulation like $R \sim \mathbb{P}$, we use $\mathbb{P}$ as a notational shorthand that summarizes the most important characteristics of R. These include a description of the values r that R may take and a recipe to compute probabilities associated with these r's. As we will see in many cases, most often we work with probability density functions that are specified by the coinciding probability distributions. In such cases, it is more convenient to think of $R \sim \mathbb{P}$ as a compact way of saying: (i) what the allowed values r of R are; and (ii) that these values obey the probability density $p(r)$ associated with $\mathbb{P}$.

Example 1.5: The normal distribution

We previously encountered the normal distribution, Normal(μ, σ^2). A shorthand like

$$R \sim \text{Normal}\left(\mu, \sigma^2\right)$$

captures the following pieces of information:

- The particular values r that R attains are real numbers ranging from $-\infty$ to $+\infty$.
- The probability density $p(r)$ of R depends on two parameters, μ and σ^2, and has the form

$$p(r) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\frac{(r-\mu)^2}{\sigma^2}\right).$$

Furthermore, the two parameters μ and σ^2 can be interpreted as the mean and the variance of R, respectively, since integration of the density leads to

$$\text{Mean of } R = \int_{-\infty}^{+\infty} dr\, r p(r) = \mu,$$
$$\text{Variance of } R = \int_{-\infty}^{+\infty} dr\, (r-\mu)^2 p(r) = \sigma^2.$$

Using the density $p(r)$, we can also compute the probability of measuring a value r between some $r_{\min}$ and $r_{\max}$. In particular, this is

$$\int_{r_{\min}}^{r_{\max}} dr\, p(r) = \frac{1}{2}\left[\text{erf}\left(\frac{r_{\max}-\mu}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{r_{\min}-\mu}{\sigma\sqrt{2}}\right)\right] \tag{1.1}$$

where erf is the error function and it is defined by an integral

$$\text{erf}\,(r) = \frac{2}{\sqrt{\pi}} \int_0^r dr'\, e^{-\frac{(r')^2}{2}}.$$

Example 1.6: The exponential distribution

The exponential distribution arises in many applications. A shorthand like

$$R \sim \text{Exponential}\,(\lambda)$$

captures the following pieces of information:

- The particular values r that R attains are real numbers ranging from 0 to ∞.
- The probability density $p(r)$ of R depends on one parameter, λ, and has the form

$$p(r) = \lambda e^{-\lambda r}.$$

The parameter λ can be interpreted as the reciprocal of the mean of R, since integration of the density leads to

$$\text{Mean of } R = \int_0^{\infty} dr\, r p(r) = \frac{1}{\lambda}.$$

Through the density $p(r)$, we can also compute the probability of measuring a value r between some $r_{\min}$ and $r_{\max}$. In particular, this is

$$\int_{r_{\min}}^{r_{\max}} dr\, p(r) = e^{-\lambda r_{\min}} - e^{-\lambda r_{\max}}. \tag{1.2}$$

Throughout this book, we extensively use several standard distributions. In examples 1.5 and 1.6 we introduced two of them though more are to come. As these will appear frequently, to refer back to them, we adopt a convention that we summarize in appendix B. Briefly, we use $R \sim \text{Normal}(\mu, \sigma^2)$ and $\text{Normal}(\mu, \sigma^2)$ to denote a normal random variable and the normal distribution, respectively. Furthermore, we use $\text{Normal}(r; \mu, \sigma^2)$ to distinguish the associated density. According to our convention, the values r of the random variable R do *not* appear in the distribution $\text{Normal}(\mu, \sigma^2)$; while, they *do* appear in the density $\text{Normal}(r; \mu, \sigma^2)$. In the latter, we separate with ";" the variable values r from the parameters μ and σ^2. We apply the same convention to the other standard distributions and densities as well.

Note 1.3: Notation

As we distinguish between a random variable R and its values r, for clarity, in this chapter we also distinguish between a probability distribution $\mathbb{P}$ and its density $p(r)$. However, in subsequent chapters, we relax this convention whenever there is no ambiguity.

Distributions on random variables with probability density functions

For a random variable R that has a probability density, we can compute the probability of attaining any of the values gathered in η, where η is a collection of r values, by the integral

$$P_\eta = \int_\eta dr\, p(r) \tag{1.3}$$

where $p(r)$ is the *probability density function* of $R \sim \mathbb{P}$ and its precise form is dictated by the distribution $\mathbb{P}$. For example, as we have seen on examples 1.5 and 1.6, a normal distribution $\text{Normal}(\mu, \sigma^2)$ has a normal density $p(r) = \exp\left(-\frac{(r-\mu)^2}{2\sigma^2}\right) / \sqrt{2\pi\sigma^2}$ and an exponential distribution $\text{Exponential}(\lambda)$ has an exponential density $p(r) = \lambda e^{-\lambda r}$. For these two, eq. (1.3) reduces to eqs. (1.1) and (1.2), respectively.

Note 1.4: Probability density over inadmissible values

By convention, the density $p(r)$ must be zero over inadmissible values r. For example, if the random variable models a distance, $p(r) = 0$ for $r < 0$; or, if the random variable models a temperature reported in absolute units, $p(r) = 0$ for $r < 0$.

By definition, the area, or more generally the volume, underneath an entire probability density $p(r)$ must be equal to 1. This is called the normalization condition and it means that an η that includes *every* admissible value r has probability 1. For instance, from eqs. (1.1) and (1.2) we can see that the probabilities of sampling any real scalar value is equal to 1 for either normal of exponential random variables.

As can be seen from eq. (1.3), a density $p(r)$ is *unitful* and its units are determined by normalization. Since $\int dr\, p(r) = 1$, where the region of integration includes every admissible value, the density $p(r)$ has the *reciprocal units* of r. So, if r is a length (in cm), the density $p(r)$ has units of reciprocal length (1/cm); or, if r is a time (in s), the density $p(r)$ has units of frequency (Hz).

Note 1.5: Re-sampling the same value of a random variable

Equation (1.3) already signals that the probability of sampling a continuous scalar random variable between some values $r_{\min}$ and $r_{\max}$ is

$$P_{r_{\min},r_{\max}} = \int_{r_{\min}}^{r_{\max}} dr\, p(r). \tag{1.4}$$

This brings up an interesting point: there is a vanishingly small probability for the same value of a continuous scalar random variable to be sampled twice with finite samplings. In fact, the probability of sampling *any particular value* is 0 as we can see by having coinciding $r_{\min}$ and $r_{\max}$ in the integral eq. (1.4). This indicates that, when thinking about continuous variables, we need to consider *intervals* of values rather than isolated values.

This feature generalizes to any continuous random variable that need not necessarily be scalar; but, as we will see shortly, it does not carry over the values of discrete random variables which can re-occur even in finite samplings. For example, a roll of 4 will re-occur multiple times in a total of 1000 rolls of a fair dice.

For a random variable R, it is also possible, and often useful, to *transform* its density $p(r)$ into a density $q(v)$ over another random variable V with values that are related by a given function $v = f(r)$. For example, such a transformation occurs when we want to apply a change to our coordinate system or simply otherwise re-parametrize our model.

Qualitatively, because we require the transformation to leave unaffected the probabilities computed using either the initial or transformed variables, the two densities must satisfy

$$\int_{\eta} dr\, p(r) = \int_{f(\eta)} dv\, q(v),$$

where $f(\eta)$ contains the transformed values $v = f(r)$ of all r in η. In the most general setting, it is hard to relate mathematically the densities $p(r)$ and $q(v)$ any further. However, provided $f(r)$ is a *differentiable* function that can be *inverted uniquely*, as is often the case in many applications, we may apply a change of variables on the right-hand-side integral to reach $\int_{\eta} dr\, p(r) = \int_{\eta} dr\, |J_{r\mapsto v}| q(v)$, where $|J_{r\mapsto v}|$ is the determinant of the transformation's Jacobian. In turn, since such an equality holds for any η, we may drop the integrals to reach a simpler form

$$q(v) = \frac{1}{|J_{r\mapsto v}|} p(r). \tag{1.5}$$

Example 1.7: Rescaling of random variables

Any physical quantity measured in real-life experiments almost always carries units. For practical reasons, often we need to convert between quantities reported in one system of units to another. Unit conversion itself is an example of variable transformation.

For concreteness, we consider a random variable R reported in some units and suppose $v = \xi r$ where v is expressed in different units from r. Here, ξ is the conversion factor, for example ξ could be 100 cm/m for v expressed in terms of centimeters and r in terms of meters. In this example, both random variables R and V are scalar, and so the Jacobian reduces to a simple derivative. More specifically, $|J_{r\mapsto v}| = |f'(r)| = \xi$ and so, the densities are

$$q(v) = \frac{p(r)}{\xi}.$$

Example 1.8: Coordinate transformation of spatial random variables

Measurements of position are reported with respect to certain frames of reference. Changing the frame of reference is another example of a variable transformation.

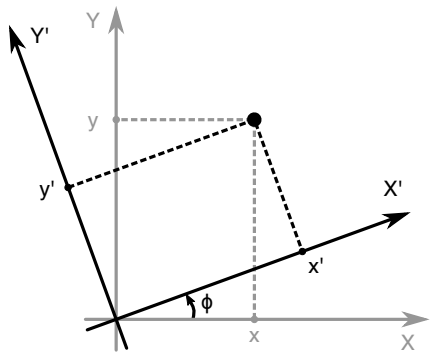


Figure 1.2: A random Cartesian position in the initial (X, Y) and the transformed (X', Y') frames of reference.

For concreteness, we consider a bivariate random variable (X, Y) that models a location in the Cartesian plane, and suppose that (X', Y') is the same location in another Cartesian frame of reference rotated through an angle ϕ about the origin, see fig. 1.2. In this case, the original and transformed positions are related through

$$x' = x\cos\phi + y\sin\phi, \qquad y' = -x\sin\phi + y\cos\phi,$$

and the Jacobian of the transformation has the form

$$J_{(x,y)\mapsto(x',y')} = \begin{bmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial y} \\ \frac{\partial y'}{\partial x} & \frac{\partial y'}{\partial y} \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}.$$

Since $|J_{(x,y)\mapsto(x',y')}| = \cos^2\phi + \sin^2\phi = 1$, the densities in the two coordinate systems are

$$q(x', y') = p(x, y).$$

Distributions on random variables with discrete values

If $\rho_{1:M} = \{\rho_1, \rho_2, \ldots, \rho_M\}$ gathers every admissible value of a *discrete* random variable R, then its probability density has the generic form

$$p(r) = \pi_{\rho_1}\delta_{\rho_1}(r) + \cdots + \pi_{\rho_M}\delta_{\rho_M}(r) = \sum_{m=1}^{M} \pi_{\rho_m}\delta_{\rho_m}(r), \tag{1.6}$$

where π_{ρ_m} are the probabilities of the individual values ρ_m contained in $\rho_{1:M}$. The terms $\delta_\rho(r)$, termed *Dirac delta*, are defined by the characteristic properties

$$\delta_\rho(r) = 0, \qquad r \neq \rho$$
$$\int dr\, \delta_\rho(r) = 1$$

where the integral is taken over every meaningful value of r; see appendix C.

Normalization, in the case of a discrete random variable, reads $1 = \sum_{m=1}^{M} \pi_{\rho_m} \int dr\, \delta_{\rho_m}(r)$, where the integral over r spans any admissible and inadmissible value. Since probabilities π_{ρ_m} are dimensionless, this implies that each $\delta_{\rho_m}(r)$ on the right hand side of eq. (1.6) has dimensions of reciprocal r, similar to the density $p(r)$. As such, normalization of a discrete random variable's density can also take the equivalent form $1 = \sum_{m=1}^{M} \pi_{\rho_m}$.

One way to represent the distribution of a random variable with a density as in eq. (1.6) is

$$R \sim \text{Categorical}_{\rho_{1:M}}(\pi_{\rho_{1:M}})$$

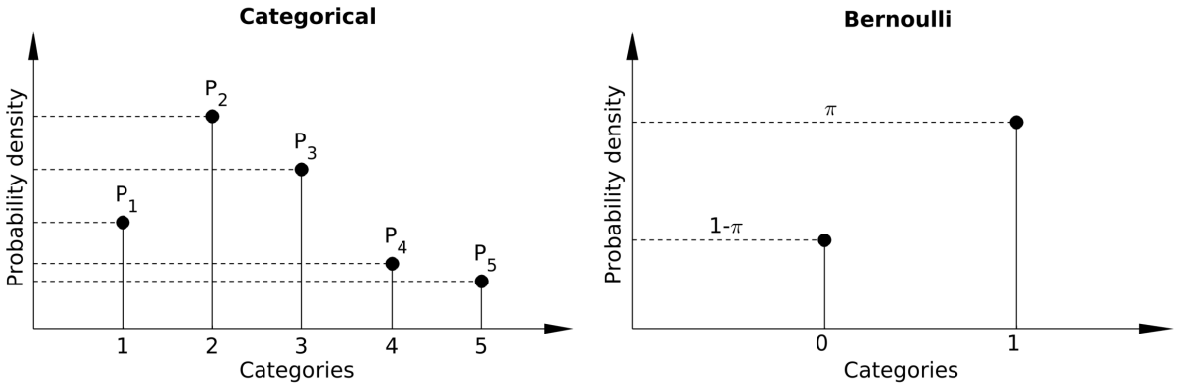


Figure 1.3: On the left hand side we plot the associated probabilities π and $1-\pi$ of the Bernoulli distribution located at its two possible outcome locations (0 and 1). On the right hand side we plot the associated probabilities $\pi_{\rho_{1:5}}$ with 5 outcomes, $\rho_{1:5}$, of the Categorical distribution.

where $\text{Categorical}_{\rho_{1:M}}(\pi_{\rho_{1:M}})$ denotes the *categorical distribution* with outcomes $\rho_{1:M}$ and associated probabilities $\pi_{\rho_{1:M}}$. A random variable drawn from this distribution samples an outcome, ρ_m, in proportion to that outcome's probability, π_{ρ_m}; see fig. 1.3.

Example 1.9: Dice rolls modeled as categorical random variables

Rolling a common dice leads to one out of six outcomes that we idealize as the faces marked with the numbers "1" through "6". Provided we identify these outcomes with the categories ρ_m, for $m = 1, \ldots, 6$, we can model a dice roll as a categorical random variable

$$R \sim \text{Categorical}_{\rho_{1:6}}(\pi_{\rho_{1:6}})$$

where the probability of face "m", or category ρ_m, is π_{ρ_m}. As we know, fair dice have equiprobable faces; $\pi_{\rho_1} = \cdots = \pi_{\rho_6} = 1/6$. However, loaded dice do not follow these probabilities.

The simplest example of a categorical distribution is the Bernoulli distribution which is the special case having just two outcomes that conventionally we identify with the numbers 1 and 0, and respective probabilities $\pi_1 = \pi$ and $\pi_2 = 1 - \pi$; see fig. 1.3. We often write $\text{Bernoulli}(\pi)$ instead of the more elaborate $\text{Categorical}_{1,0}(\pi, 1-\pi)$.

Example 1.10: Coin flips modeled as Bernoulli random variables

An ideal coin flip has only two outcomes: "heads" or "tails". Provided we identify these with the numbers 1 and 0, respectively, we can model a coin flip as a Bernoulli random variable

$$R \sim \text{Bernoulli}(\pi) \tag{1.7}$$

where π is the probability of "heads". Here, specifying the probability of tails, $1 - \pi$, is redundant since, by normalization, it is uniquely determined by π.

If, instead, we want to avoid identifying "heads" and "tails" with 1 and 0, we can also model a coin flip as a categorical random variable

$$R \sim \text{Categorical}_{\text{heads,tails}}(\pi, 1-\pi). \tag{1.8}$$

Essentially, the only difference between eq. (1.7) and eq. (1.8) is in the meaning we assign to the values r, with the latter representation here having an *interpretational advantage* over the former.

Distributions on random variables *without* probability density functions*

Since in later chapters we formulate models with random variables to which we *cannot* assign a probability density function, for example random variables that are functions or random variables that are probability distributions themselves, we also need to account for appropriate distributions on those. In such cases recipes to compute probabilities are case specific and, in general, the description of the associated distributions is considerably more complicated. In examples 1.11 and 1.12 we provide only a sneak preview.

Example 1.11: The standard Brownian motion

We will examine Brownian motion in more detail in chapter 2. As we will see, standard Brownian motions in 1D are random variables that represent functions from a time interval spanning 0 to some positive T to the real line. To denote them we write

$$X \sim \text{BMotion}_T^{\text{1D}}\left(D\right)$$

where the parameter D in the Brownian motion is a positive real scalar and, as we will see, can be interpreted as a diffusion coefficient of a particle diffusing in 1D.

A shorthand like this captures the following pieces of information:

- The realizations of X are functions $x(\cdot)$ that, to any time t between 0 and T, assign $x(t)$ which is a position on the real line.
- Any realization of X, is initialized at the origin, *i.e.* $x(0) = 0$.
- For any choice of times t and t' between 0 and T, the difference $x(t) - x(t')$ between the values $x(t)$ and $x(t')$ of any realization $x(\cdot)$ is a random variable itself.
- The random variable $x(t) - x(t')$ has a probability density given by

$$p\left(x(t) - x(t')\right) = \frac{1}{\sqrt{4\pi D|t-t'|}} \exp\left(-\frac{\left(x(t) - x(t')\right)^2}{4D|t-t'|}\right).$$

Example 1.12: The Gaussian process

We will examine *Gaussian processes* in more detail in section 7.1. As we will see, Gaussian processes are random variables that represent functions from a space S to the real numbers. To denote them we will write

$$F \sim \text{GaussianP}_S\left(h(\cdot), c(\cdot,\cdot)\right).$$

A shorthand like this captures the following pieces of information:

- The realizations of F are functions $f(\cdot)$ that, to any point x in S, assign $f(x)$ which is a real number.
- The parameter $h(\cdot)$ is a function that, to every point x in S, assigns $h(x)$ which is also a real number.
- The parameter $c(\cdot,\cdot)$ is a function that, to every points x and x' in S, assigns $c(x,x')$ which is a non-negative real number.
- For any choice $x_1, \ldots, x_M$ of any finite number M of points in S, the values $[f(x_1), \cdots, f(x_M)]$ form a random array.
- The random array $[f(x_1), \ldots, f(x_M)]$ has a probability density

$$p\left([f(x_1), \cdots, f(x_M)]\right) = \text{Normal}_M\left(\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_M) \end{bmatrix}; \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_M) \end{bmatrix}, \begin{bmatrix} c(x_1,x_1) & \cdots & c(x_1,x_M) \\ \vdots & \ddots & \vdots \\ c(x_M,x_1) & \cdots & c(x_M,x_M) \end{bmatrix}\right).$$

*This is an advanced topic and could be skipped on a first reading.

1.2.2 How to simulate probability distributions

So far we have discussed random variables and probability distributions from which random variables are drawn. What we discuss next is how to run *simulations*. That is, how to sample random variables in a computer using their probability distributions in order to re-create *in silico* repetitions of our prototype experiment. In subsequent chapters, we will see that we can use sampling not only to re-create an experiment's results but also to draw inferences *from* an experiment's results.

Continuous random variables

For a random variable $R \sim \mathbb{P}$ that takes scalar real values r, its *probability cumulative function* is a function $C(r)$ given by

$$C(r) = \int_{-\infty}^{r} dr'\, p(r') \tag{1.9}$$

where $p(r)$ is the probability density associated with $\mathbb{P}$. From this definition, we see that a cumulative function is dimensionless and increases monotonically between 0 to 1. This is a characteristic that we can use to develop a method from which to sample r on a computer.

For instance, given a density $p(r)$, we first calculate its cumulative function according to eq. (1.9). We then generate a random value, call it u, uniformly between 0 and 1, and ask: *for what value r is the cumulative function equal to u?* In other words, we find $r = C^{-1}(u)$, where $C^{-1}(u)$ is the inverse function of $C(r)$. This method, often termed the *fundamental theorem of simulation*, is summarized in algorithm 1.1 and is visually illustrated in fig. 1.4.

Algorithm 1.1: Fundamental theorem of simulation for continuous variables

To simulate a continuous random variable $R \sim \mathbb{P}$

- First, find the cumulative function $C(r)$ and its inverse $C^{-1}(u)$.
- Then, repeat the following steps
  - Generate $u \sim \text{Uniform}_{[0,1]}$.
  - Set $r = C^{-1}(u)$.

Upon completion, this algorithm draws values r according to $\mathbb{P}$.

Example 1.13: Simulating from an exponential distribution

Consider an exponential random variable $R \sim \text{Exponential}(\lambda)$. As we saw in example 1.6, this random variable takes real scalar values and its density is

$$p(r) = \begin{cases} \lambda e^{-\lambda r}, & r \geq 0 \\ 0, & r < 0 \end{cases}.$$

To apply the fundamental theorem of simulation, we first compute the cumulative function and its inverse. These are

$$C(r) = 1 - e^{-\lambda r}, \qquad C^{-1}(u) = -\frac{1}{\lambda}\log(1-u).$$

By means of algorithm 1.1, we then sample R as follows:

- First, we generate u from a uniform distribution between 0 and 1

$$u \sim \text{Uniform}_{[0,1]}.$$

- Then, we compute r from

$$r = -\frac{1}{\lambda}\log(1-u). \tag{1.10}$$

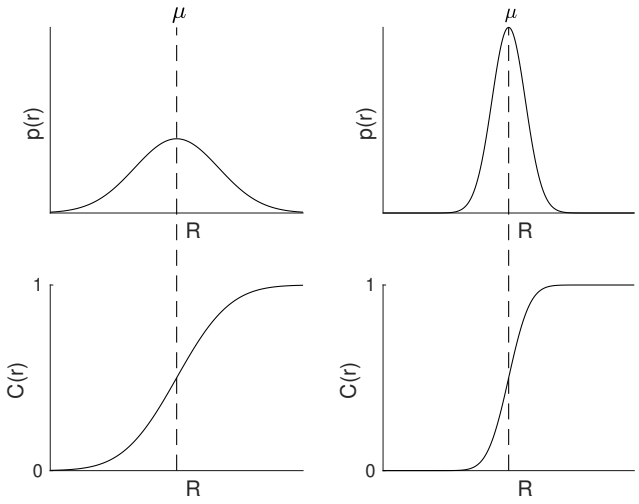


Figure 1.4: On the top rows we have PDFs broadly and tightly centered around their mean μ. In the bottom row, we have the coinciding CDF. The CDF has a sharp slope near $r = \mu$ for the PDF more tightly centered on its mean. Thus, for this special case in applying algorithm 1.1, most values of u would coincide with values of r near μ. By contrast, if the PDF were broader near μ, the slope of the CDF would be smaller and values of u would coincide with a broader range of values of r.

Since $v = 1 - u$ is also uniformly distributed between 0 and 1, for computational efficiency, when sampling exponential random variables, we generate $v \sim \text{Uniform}_{[0,1]}$ in the first place and then use $r = -\frac{1}{\lambda} \log v$ instead of eq. (1.10). In this way, we speed up the execution of the algorithm by avoiding the computation of the difference $1 - u$.

Note 1.6: Nomenclature

So far, we have encountered three important functions $p(r), C(r), C^{-1}(u)$ associated with a random variable $R \sim \mathbb{P}$. These are very common in the literature and below we summarize some terms that are used to designate them.

- $p(r)$ is occasionally termed *probability density function* or *PDF*.
- $C(r)$ is occasionally termed *probability cumulative function, cumulative distribution function* or *CDF* .
- $C^{-1}(u)$ is occasionally termed *probability quantile function, inverse cumulative distribution function* or *ICDF* .

Discrete random variables

We can use a similar procedure to sample discrete random variables too. In particular, for $R \sim \text{Categorical}_{\rho_{1:M}}(\pi_{\rho_{1:M}})$, the cumulative function is

$$C(\rho_m) = \sum_{m'=1}^{m} \pi_{\rho_{m'}}$$

or, more concretely, it has the form

$$C(\rho_1) = \pi_{\rho_1}, \qquad C(\rho_2) = \pi_{\rho_1} + \pi_{\rho_2}, \qquad \cdots \qquad C(\rho_M) = \pi_{\rho_1} + \pi_{\rho_1} + \cdots + \pi_{\rho_M}.$$

To sample an outcome r, as with continuous random variables, we also need to generate $u \sim \text{Uniform}_{[0,1]}$. However, now a problem concerning the inversion of $C(r)$ arises. Namely, there may be no r such that $C(r) = u$.

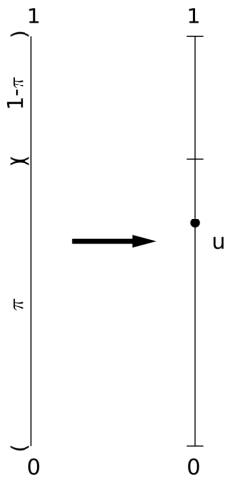


Figure 1.5: On top we consider a probability, ranging from 0 to 1, segmented into two portions of weight π and $1-\pi$ separated by a break point. We imagine these to be the probability of sampling outcome 0 or outcome 1 in a Bernoulli trial. To determine which outcome we select, we draw a uniform random number u. In this figure, the u sampled fell below the break point. As such, we select outcome 0 for this Bernoulli trial.

For this reason, instead of searching for outcomes such that $C(r) = u$, we search for the *lowest* value r such that $u \leq C(r)$. This version of the fundamental theorem of simulation is summarized in algorithm 1.3.

Algorithm 1.2: Fundamental theorem of simulation for discrete variables

To simulate a random variable $R \sim \text{Categorical}_{\rho_{1:M}}(\pi_{\rho_{1:M}})$

- Generate $u \sim \text{Uniform}_{[0,1]}$.
- Find the lowest m such that $u \leq \pi_{\rho_1} + \pi_{\rho_2} + \cdots + \pi_{\rho_m}$.
- Set $r = \rho_m$.

At first, it might appear that this algorithm depends on the particular labeling of $\rho_{1:M}$ and that it would lead to different realizations r if the labels m had been assigned differently over the categories ρ_m. However, since relabeling of $\rho_{1:M}$ involves also a similar relabeling of $\pi_{\rho_{1:M}}$, this is *not* the case. In other words, this algorithm realizes each outcome $r = \rho_m$ with probability π_{ρ_m}, even when the labels m are reassigned over ρ_m.

Example 1.14: Simulation for Bernoulli random variables

Consider $R \sim \text{Bernoulli}(\pi)$. In this case, the cumulative function has a very simple form

$$C(1) = \pi, \qquad C(0) = 1.$$

To sample r, according to the fundamental theorem of simulation

- first, we generate $u \sim \text{Uniform}_{[0,1]}$.
- then, if $u \leq \pi$ we set $r = 1$, else we set $r = 0$.

The two steps are illustrated in fig. 1.5.

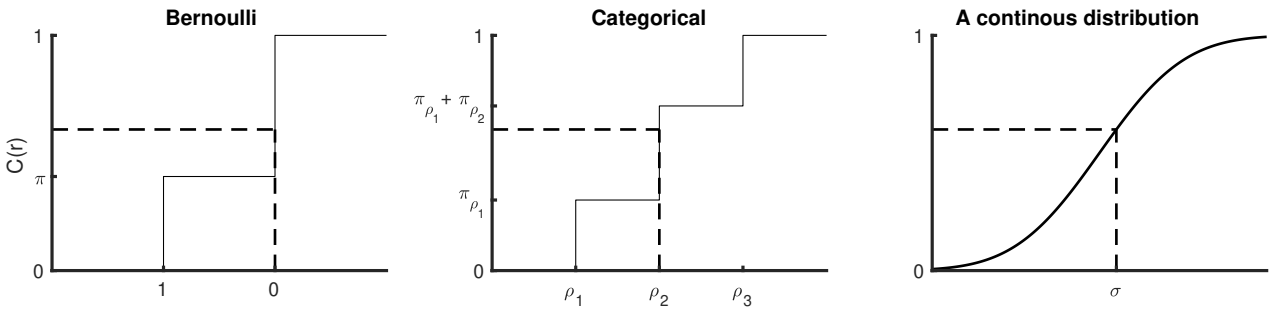


Figure 1.6: The continuous distribution can be thought of as a Categorical distribution in the limit that a continuum number of realizations of the random variable are allowed. As such, the value at the abscissa at which the horizontal line u intersects with the CDF dictates which random variable is realized.

Note 1.7: How the fundamental theorems of simulation work

There exists an intuitive explanation for the fundamental theorem of simulation that we illustrate in fig. 1.6. That is, we imagine a stick of unit length with a break point just as shown in fig. 1.5. The portion of the stick before the break point has length π. The remainder of the stick has length $1 - \pi$. We now sample a uniform random variable, u. If u falls before the break point, outcome 0 is realized. Otherwise outcome 1 is realized.

A similar logic holds for the Categorical distribution. Figure 1.6 shows the discrete steps in the cumulative function of a discrete distribution. A draw from the uniform distribution can be visualized as the dotted horizontal line of fig. 1.6. The value of the abscissa that coincides with the location where the dotted line intersects with the CDF dictates the value realized by the discrete random variable.

The next logical leap we need to take is to think of a continuous distribution as the limit of a Categorical with very many closely packed realizations of the random variable. The reasoning underlying how we go about sampling from a continuous distribution then follows from the argument put forward for sampling from a Categorical. This is shown in fig. 1.6.

1.2.3 How to combine probability distributions

In a complex model, we most likely have multiple random variables and, generally, to each one is attributed its own properties. This means that each random variable follows its own distribution and so, when handling a complex model, we work with more than one distribution. Here, we present rules to combine and manipulate distributions. As we will see, such rules are used regularly as they greatly facilitate bookkeeping.

Joint and marginal distributions

Provided the random variables are independent from each other, for example because they may model physical processes or observations that exert no influence upon each other, we may write

$$R_1 \sim \mathbb{P}_1, \qquad R_2 \sim \mathbb{P}_2, \qquad \cdots \qquad R_N \sim \mathbb{P}_N.$$

Each distribution $\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_N$ is, in turn, associated with its own density $p_1(r_1), p_2(r_2), \ldots, p_N(r_N)$. As there is little chance of confusion, commonly we simply write $p(r_n)$ instead of $p_n(r_n)$.

Occasionally, we also encounter models with multiple random variables

$$R_1 \sim \mathbb{P}, \qquad R_2 \sim \mathbb{P}, \qquad \cdots \qquad R_N \sim \mathbb{P}, \tag{1.11}$$

that are independent and which also follow identical distributions $\mathbb{P}$, for example random variables that may model independent observations obtained from a time invariant system. On such occasions, we might abbreviate

eq. (1.11) into $R_1, R_2, \ldots, R_N \overset{iid}{\sim} \mathbb{P}$ and speak of *independent and identically distributed*, or simply *iid*, random variables. Essentially, we mean that all densities $p(r_1), p(r_2), \ldots, p(r_N)$ happen to have the same form. In the iid setting, and only when there is no chance of confusion, we might refer to each one of the densities simply as $p(r)$. However, as we will see shortly, even with iid variables, most of the times we run into complicated settings and what we mean by $p(r)$ needs explicit clarification.

Following our convention, we denote the density of a single variable R_n as $p(r_n)$ and call it a *marginal density*. When multiple random variables $R_{1:N}$ arise in the same setting and a density depends on all of them, we write $p(r_{1:N}) = p(\{r_1, r_2, \ldots, r_N\})$ or $p(r_{1:N}) = p(r_1, r_2, \ldots, r_N)$ and refer to $p(r_{1:N})$ as a *joint density*.

A *marginal density* $p(r_n)$ can be obtained from the joint density $p(r_{1:N})$ through an integration over the entire range spanned by $r_{1:n-1}$ and $r_{n+1:N}$. That is,

$$p(r_n) = \underbrace{\int dr_1 \cdots \int dr_{n-1} \int dr_{n+1} \cdots \int dr_N}_{\text{everything but } r_n} \; p(r_{1:N}). \tag{1.12}$$

Colloquially, we refer to the integration over variables, *i.e.* going from right-to-left in eq. (1.12), as a "marginalization". We refer to the reverse process, *i.e.* going from left-to-right in eq. (1.12), as a "de-marginalization" or a "completion".

Example 1.15: Marginalization over probability densities

Following the same logic, we may obtain distributions over any subset of the variables in $R_{1:N}$. For concreteness, we consider a total of $N = 5$ variables and suppose that we wish to obtain a distribution over R_2 and R_4 only. In this case, marginal and joint densities are linked by

$$p(r_2, r_4) = \underbrace{\int dr_1 \int dr_3 \int dr_5}_{\text{everything but } r_2 \text{ and } r_4} \; p(r_{1:5}).$$

Note 1.8: Box-Muller simulation of normal random variables

We are now ready to discuss the simulation of a random variable from the normal distribution $X \sim \text{Normal}(\mu, \sigma^2)$. As the cumulative function of X does not have a closed form and we cannot use the fundamental theorem of simulation, we follow a different approach that relies on joint distributions.

We start by considering two iid random variables, X, Y. In particular,

$$X, Y \sim \text{Normal}(\mu, \sigma^2).$$

The associated joint density reads

$$p(x, y) = p(x)p(y) = \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}} \right) \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \right).$$

On X and Y, as we illustrate on fig. 1.7, we perform three successive transformations:

- a linear transformation from x and y to

$$x' = \frac{x - \mu}{\sigma}, \qquad y' = \frac{y - \mu}{\sigma}$$

- a non-linear transformation from Cartesian (x', y') to polar coordinates (ρ, ϕ) with

$$x' = \rho \cos\phi, \qquad y' = \rho \sin\phi$$

- a non-linear transformation from ρ to λ with

$$\lambda = \rho^2.$$

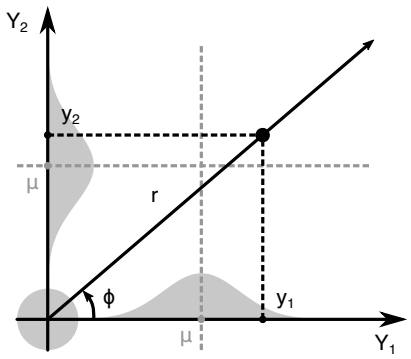


Figure 1.7: Sampling independent random variables $X, Y \sim$ Normal(μ, σ^2) according to the Box-Muller method of note 1.8 by, equivalently, sampling an r and ϕ in polar coordinates.

The advantage of applying these transformations is that the resulting density over λ and ϕ is separable. In particular

$$p(\lambda, \phi) = \frac{1}{2}\exp\left(-\frac{\lambda}{2}\right)\frac{1}{2\pi} = \text{Exponential}\left(\lambda; \frac{1}{2}\right)\text{Uniform}_{[0,2\pi]}(\phi).$$

The cumulative function over λ and ϕ can now be computed analytically. So, by generating two uniform random samples $u_1, u_2 \sim \text{Uniform}_{[0,1]}$, we can readily obtain random samples from the radial and polar angle distribution

$$\rho = \sqrt{\lambda} = \sqrt{-2\log u_1}, \qquad \phi = 2\pi u_2.$$

Transforming back to our original variables, we obtain

$$x = \mu + \sigma x' = \mu + \sigma\rho\cos\phi = \mu + \sigma\sqrt{-2\log u_1}\cos(2\pi u_2).$$

This algorithm for sampling normal random variables is termed the *Box-Muller method*. As can be seen, with little additional computational cost, this method also provides another normal sample

$$y = \mu + \sigma\sqrt{-2\log u_1}\sin(2\pi u_2)$$

which is independent of x.

Conditional distributions

The order in which random variables arise in a model may be irrelevant, for example random variables modeling an experiment's observations that exert no influence upon each other, such as individual test scores or measurements of heights collected from a group of unrelated adults. On the other hand, the order in which random variables arise *may be important*, for example random variables modeling observations of time-dependent phenomena, such as successive measurements of the number of cells in a growing cell culture or the number of molecules available to react in a chain of chemical reactions.

To express *dependencies* among two random variables R_1 and R_2 we write

$$R_2|r_1 \sim \mathbb{P}(r_1). \tag{1.13}$$

This reads "the random variable R_2, given the realization r_1 of the random variable R_1, is sampled from the probability distribution $\mathbb{P}(r_1)$" and means that the values of r_2 are associated with a density $p(r_2|r_1)$ that depends upon r_1. We designate a distribution that depends upon the value of another random variable like $\mathbb{P}(r_1)$ and the associated density $p(r_2|r_1)$ as *conditionals*.

Note 1.9: How to avoid inaccuracies in specifying variable dependencies

In the setting of eq. (1.13), the random variable R_1 is sampled from its own (marginal) distribution that needs to be specified *separately*. In a complete model, both random variables $R_1 \sim \mathbb{P}_1$ and $R_2|r_1 \sim \mathbb{P}_2(r_1)$ need to be specified adequately.

We can never sufficiently overemphasize that, in a properly formulated model, the distribution of R_1 *must not* depend upon r_2 and, for this reason, the description of R_1 should precede that of $R_2|r_1$. If in certain models, this is not possible, then the two random variables should be described together through a joint distribution $(R_1, R_2) \sim \mathbb{P}$.

Ideally, proper descriptions involving multiple random variables, that depend upon each other, should be given in a nested fashion. For example

$$R_1 \sim \mathbb{P}_1$$
$$R_2|r_1 \sim \mathbb{P}_2(r_1)$$
$$R_3|r_2, r_1 \sim \mathbb{P}_3(r_2, r_1)$$
etc...

A necessary condition (although not always sufficient) for a reliable description of a probabilistic model, no matter how convincing the involved arguments may be and no matter how intuitive the involved distributions may appear, is that *every single distribution* $\mathbb{P}_1, \mathbb{P}_2(r_1), \mathbb{P}_3(r_2, r_1), \ldots$ be specified *clearly and explicitly*.

Whenever a supposedly flawless model cannot be put in a nested form as above, even when random variables are grouped and joint distributions are applied, the model most likely contains flaws such as tautologies or contradictions. Consequently, such a model is inappropriate for quantitative or even qualitative use.

In note 1.9, we consider nested variable dependencies. In particular, R_3 depends on the realizations r_2 and r_1, in turn, R_2 depends on the realization r_1, and finally R_1 depends on no other realization. In the most general case, the probability distribution over the last random variable, say R_K, may depend on the realization of all previous random variables, $r_{1:K-1}$, and the same happens for all other variables up to the very first one r_1. Because of this hierarchy, to simulate a nested model we may use a sampling algorithm termed *ancestral sampling* as detailed below.

Algorithm 1.3: Ancestral sampling

To draw values for $R_{1:K}$, we proceed as follows:
- Find the density $p(r_1)$ associated with $R_1 \sim \mathbb{P}_1$.
- Sample r_1 using $p(r_1)$.
- for k from 2 up to K, repeat:
  - Find the density $p(r_k|r_{1:k-1})$ associated with $R_k|r_{1:k-1} \sim \mathbb{P}_k(r_{1:k-1})$.
  - Sample r_k using $p(r_k|r_{1:k-1})$.

Because ancestral sampling and hierarchal models are very common, we now describe methods to obtain the necessary conditional densities. Our starting point is the full joint density $p(r_{1:K})$.

Conditional and joint densities are related to each other through the *chain rule* which, in the most general setting, reads

$$p(r_{K:1}) = p(r_K|r_{K-1:1}) \cdots p(r_2|r_1)p(r_1).$$

In the simplest case consisting of only two random variables, the chain rule reads

$$p(r_2, r_1) = p(r_2|r_1)p(r_1).$$

From this we immediately see that a conditional density over r_2 is normalized over r_2 irrespective of r_1, *i.e.*

$$\int dr_2\, p(r_2|r_1) = \int dr_2\, \frac{p(r_2, r_1)}{p(r_1)} = \frac{\int dr_2\, p(r_2, r_1)}{p(r_1)} = \frac{p(r_1)}{p(r_1)} = 1.$$

Additionally, from the chain rule, we obtain two equalities, $p(r_2, r_1) = p(r_2|r_1)p(r_1)$ and $p(r_1, r_2) = p(r_1|r_2)p(r_2)$, that we can combine to obtain another important rule, namely *Bayes' rule*, which most often is written in the form

$$p(r_2|r_1) = \frac{p(r_1|r_2)p(r_2)}{p(r_1)}, \qquad p(r_1) \neq 0. \tag{1.14}$$

As we will see in subsequent chapters, eq. (1.14) is an indispensable tool in Data Analysis.

Note 1.10: Modeling dynamical systems

Dependency among variables is especially important when the physical system of interest evolves over time. In this dynamical setting, which we explore in the next chapter, our prototype experiment is temporally structured: causality indicates that the last measurement W_N may be influenced by all preceding measured values, $w_{1:N-1}$; the penultimate measurement, W_{N-1}, may be influenced by all of its preceding ones $w_{1:N-2}$; and so forth.

With the rules of joint and conditional distributions, we can work out the densities of such models in the most general setting. For instance,

$$p(w_{1:N}) = p(w_N|w_{1:N-1})p(w_{N-1}|w_{1:N-2}) \cdots p(w_2|w_1)p(w_1).$$

It follows that if we need to sample realizations of $W_{1:N}$, we need 1 marginal and $N-1$ *different* conditional distributions. As a result, this sampling may become infeasible unless we make some assumptions.

- One drastic assumption, often too crude for realistic dynamical systems, is to assume that all variables are independent, which in this particular case is equivalent to assuming $p(w_n|w_{1:n-1}) = p(w_n)$. Under this assumption, the joint density factorizes into a product of densities

$$p(w_{1:N}) = p(w_1)p(w_2) \cdots p(w_N) = \prod_{n=1}^{N} p(w_n). \tag{1.15}$$

- Another, less drastic, and often realistic, assumption is to consider $p(w_n|w_{1:n-1}) = p(w_n|w_{n-1})$. Under this assumption, the joint density also factorizes into a product of densities

$$p(w_{1:N}) = p(w_1)p(w_2|w_1) \cdots p(w_N|w_{N-1}) = p(w_1) \prod_{n=2}^{N} p(w_n|w_{n-1}). \tag{1.16}$$

Under these two assumptions, the total number of different probability distributions, that are needed to sample $W_{1:N}$, reduces from N to 1 and 2, respectively.

Somewhat pedantically, in deriving eq. (1.15), we invoked a so-called *0th order Markov assumption*

$$p(w_n|w_{1:n-1}) = p(w_n),$$

while, in deriving eq. (1.16) we invoked a so-called *1st order Markov assumption*

$$p(w_n|w_{1:n-1}) = p(w_n|w_{n-1}).$$

In principle, we can also invoke higher order assumptions where a measurement w_n is influenced by more than 1 past measurement; however, as we will see in the subsequent chapters, such assumptions are rarely used in practice, either because a 1st order assumption is already sufficient or because they lead to models with prohibitive computational cost.

1.3 Data-driven modeling and inference

Our emphasis, throughout this book, is not as much on mathematical rigor as it is focused on problem-formulation and problem-solving. But, *of what problem exactly?*

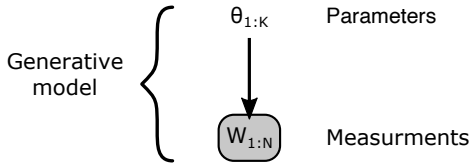


Figure 1.8: A generative model describes how measurements are generated. Implicitly, it encodes any influence the parameters $\theta_{1:K}$ exert upon the measurements $W_{1:N}$.

In the data-centric context that is most appropriate for the Physical and Natural Sciences, we envision being provided information on a physical system such as:

- *how this system behaves* under relevant, well or poorly characterized, conditions;
- *how observations are acquired* on this system;
- *specific values* of acquired observations.

These are the *data* and they serve as our input or starting point. Our primary task is to analyze the data and we tackle *Data Analysis* with the framework introduced in section 1.1.2. More specifically, within the framework set by the prototype experiment, which we specialize to particular scenarios, our goal is to use the acquired values of the observations to infer a model. However, before we can infer a model, we need to go through a *synthesis stage* to develop the necessary mathematical formulation that is meaningful for the system at hand.

First, we utilize the available information on our system to formulate the probability distribution $p(w_{1:N}|\theta_{1:K})$ describing our experiment. For example, in this stage we consider physical laws, intrinsic dynamics, and noise properties, which, although non-numeric, in a very concrete sense are part of our given data. At this stage, we also decide on parameters $\theta_{1:K}$ and probably assign a physical meaning to all or some of them. This stage concludes with the establishment of a *generative model*; that is, a quantitive description of *how our experiment's measurements are generated (sampled)*, see fig. 1.8, and in principle could be simulated in a computer.

Note 1.11: A model's likelihood

The probability distribution $p(w_{1:N}|\theta_{1:K})$, which is established in a generative model, is a key quantity. This distribution is termed the *likelihood*. The term follows from the notion that $p(w_{1:N}|\theta_{1:K})$ quantifies the likelihood of observing (sampling) the sequence of observations $w_{1:N}$ in our prototype experiment which is influenced by the parameters $\theta_{1:K}$.

Second, once we formulate $p(w_{1:N}|\theta_{1:K})$ adequately, we apply the measured values of $w_{1:N}$ to compute specific estimates of the parameters $\theta_{1:K}$. Commonly, we call these values *estimators* and denote them with $\hat{\theta}_{1:K}$.

Note 1.12: Likelihood based estimation

A likelihood provides us with a *universal* strategy to estimate parameters $\hat{\theta}_{1:K}$ needed to specify uniquely a model we wish to learn. The challenge, however, is that we are also often interested in error bars around $\hat{\theta}_{1:K}$ or, put differently, whole probability distributions over $\theta_{1:K}$. For this reason, in chapter 4, we will consider an extended strategy that uses more than an experiment's likelihood.

The first stage takes more of a *modeling perspective;* while, the second stage takes more of a *computational perspective* which. Nonetheless, as we discuss in example 1.16, both stages in the solution of our problem are important and both stages pose unique challenges. In practice, as we will see in subsequent chapters, often we have to devise comprehensive approaches that deal with the challenges arising in both stages simultaneously.

Example 1.16: Likelihood based modeling and inference

As a concrete example, we imagine an experiment idealized as having one of two (discrete) measurement outcomes, for example the emission of a photo-electron or no emission as given in example 1.3. For simplicity, we may denote these outcomes with $\rho_1 = 1$ and $\rho_2 = 0$, respectively.

If we idealize individual assessments as iid, meaning that each measurement is independent of the others as in eq. (1.15), then the mathematical form of the likelihood is readily derived. In particular, the model responsible for generating the data takes the form

$$W_n|\pi \sim \text{Bernoulli}(\pi), \qquad n = 1, \ldots, N,$$

and, as of yet, has one unspecified parameter, namely π, which is the probability that a single assessment measures a photo-electron. Our goal is therefore to estimate π.

Now we ask: Given this *generative* model what is the *likelihood* of our measurements? This likelihood is the probability of observing the sequence $w_{1:N}$ and we may compute it as following

$$p(w_{1:N}|\pi) = \prod_{n=1}^{N} p(w_n|\pi) = \prod_{n=1}^{N} \text{Bernoulli}(w_n;\pi) = \prod_{n=1}^{N} \pi^{w_n}(1-\pi)^{1-w_n} = \pi^M(1-\pi)^{N-M}$$

where we assumed that, within $w_{1:N}$, the first outcome, ρ_1, has been observed in total M times and the second outcome, ρ_2, has been observed the remainder of the times, $N - M$.

We can estimate a value for the parameter π by asking: *Which value of π makes our observations most likely?* This is equivalent to asking which value of π makes $p(w_{1:N}|\pi)$ highest. Essentially, we need to seek for the maximizer of $p(w_{1:N}|\pi)$. For instance, solving $\frac{d}{d\pi}p(w_{1:N}|\pi) = 0$, we find $\hat{\pi} = M/N$, as expected intuitively.

For this example, we assumed that both outcomes, ρ_1 and ρ_2, are observed at least once and so $0 < \hat{\pi} < 1$, because $0 < M < N$. Yet had $M = 0$ or $M = N$, we may had erroneously concluded, due to limited data, that $\hat{\pi} = 0$ or $\hat{\pi} = 1$. In turn, if we had used this estimate to predict the outcome of future experiments, we would had erroneously concluded that this would *conclusively* be either ρ_2 or ρ_1, respectively. Thus, even this toy example forebodes our need to go beyond approaches that rely exclusively on likelihoods.

In the Physical Sciences, data-driven approaches are sometimes termed *inverse methods*, *inverse problems*, or *inverse modeling*. Yet, as example 1.16 illustrates, there is nothing backward about obtaining models starting from the data and these, somewhat unfortunate terms, arose only because traditional approaches–that is, obtaining models from the ground-up with a combination of first principles and data-fitting–came historically first and are now termed forward (or direct).

Note 1.13: Inverse modeling

Data-driven model inference essentially is an inverse problem. Solving an inverse problem is the opposite of solving a direct problem. Briefly, in a *direct problem* we seek to determine an effect knowing its cause; while, in an *inverse problem* we seek to recover the cause knowing only the effect.

Inverse problems arise mainly when we need to interpret indirect physical measurements of unknown or partially known origin. For instance, when we are interested in elucidating the dynamics of complex biomolecules observed indirectly through fluorescence microscopy. In an experiment we acquire images (measurements) with all sorts of artifacts that subsequently need to be removed in order to reveal the positions or dynamics of the biomolecules we are interested in. By contrast, simulating possible measurements invoking a physical model, established or tentative, that *predicts* the positions or dynamics sought after and subsequently checking whether they are in agreement or disagreement with the observed measurements is forward modeling and essentially involves solving only direct problems; see fig. 1.9.

A problem, whether direct or inverse, is *well-posed* when it meets the following conditions

- the problem has a solution;

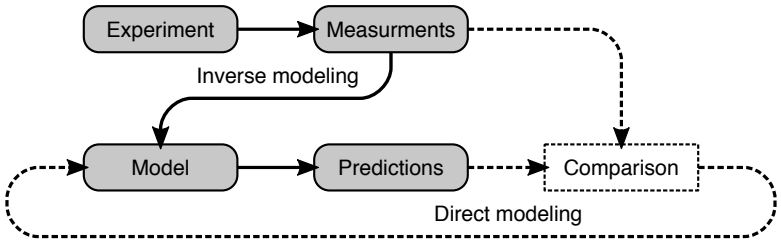


Figure 1.9: Illustration of the direct and inverse modeling paradigms. In the direct paradigm, a model is adjusted until its predictions agree with an experiment's measurements. By contrast, in the inverse paradigm, a model is inferred from experimental measurements with no adjustments.

- the solution is unique;
- the solution does not differ substantially unless the supplied data differ substantially too.

These conditions are known as *existence, uniqueness,* and *stability*, respectively. If a problem fails to satisfy one or more of them, it is *ill-posed*.

Direct problems are well-posed when the effects (data) we are after are well-defined, single-valued, and depend continuously on their causes. Often this is the case when we seek to reproduce observations mathematically or computationally. On the other hand, solutions to inverse problems do *not always exist*, or when they exist they *may not be unique* or they may *change dramatically* even when the supplied data (effects) differ only insignificantly. As a result, inverse problems are commonly ill-posed and solving them can be very challenging.

Throughout the subsequent chapters, with the use of appropriate random variables and probability distributions, we will see how inverse problems can be re-formulated as proper statistical problems and how solutions to these problems can be computed robustly and efficiently.

Forward modeling has had its role to play and is heavily showcased throughout Physics where disparate observations were unified into predictive frameworks inspired by logic, symmetries and fundamental postulates. Undoubtedly, the forward approach has been tremendously successful. To wit, among others, it predicted the magnetic moment of the electron to a spectacular number of significant digits. But there are limitations to this historically successful approach.

While forward modeling historically came first, inverse methods, spurred in equal parts by advances in probability theory and motivating data-centric questions in the Natural Sciences, also arose. Today, large swathes of complicated enough physical and chemical systems, in addition to Life and Social Sciences, are not naturally modeled from the ground-up, *i.e.* starting from first principles. Instead, in these cases, observations often only suggest loose couplings between variables of interest and probabilistic relations between various quantities that must be aided by data.

The forward approach is different from the philosophy we adopt here. Instead, we use the first principles only to motivate forms for our generative models. Beyond this, we are motivated by the practice of Statistics that instead attempts, from the onset, to be as agnostic about the model parameters (or the model itself) as possible and learn parameters and models self-consistently from the available data as efficiently as computationally possible.

1.4 Exercise problems

Exercise 1.1: Math warm-up

Evaluate the following by hand.

1. Gaussian integral: $\int_{-\infty}^{+\infty} dx\, e^{-(x-\mu)^2/(2\sigma^2)}$, assume μ and σ^2 are real scalars and $\sigma^2 > 0$.
2. Gaussian moments: $\int_{-\infty}^{+\infty} dx\, x^2 e^{-(x-\mu)^2/(2\sigma^2)}$, assume μ and σ^2 are real scalars and $\sigma^2 > 0$
3. Gaussian product: Show that the product of N Gaussians remains a Gaussian.
4. Gamma-function integral: Assume n is a positive integer and show that $\int_0^\infty dx\, x^n e^{-x}$ is equal to $n!$.
5. Gamma-function integral: Assume n is positive integer, a is a positive real scalar and show that $\int_0^\infty dx\, x^n e^{-x/a}$ is equal to $a^{n+1} n!$.
6. Geometric series: $\sum_{n=0}^{N} x^n$, assume N is a positive integer.
7. Poisson variance: $\sum_{n=0}^{\infty} n^2 \lambda^n \exp(-\lambda)/n!$, assume λ is real and positive.
8. Taylor expand in x, around 0, to second order the following:

$$\frac{1}{1-\frac{x}{a}}, \qquad \frac{e^x}{1-x}, \qquad \frac{e^x \log(1-x^2)}{1-x}, \qquad \sqrt{1-2x},$$

assuming a is a constant real scalar.

Exercise 1.2: Permutations and combinations

Consider integers $N = 1, 2, \ldots$ and $M = 0, 1, \ldots, N$.

1. Show that the total number of distinct arrangements (permutations) of M objects selected out of N distinct objects is $\frac{N!}{(N-M)!}$.
2. Show that if we ignore the arrangement of the objects (combinations) the total number drops to $\frac{N!}{M!(N-M)!}$.
3. Show that the total number of different combinations of N distinct objects is 2^N.

Exercise 1.3: Cumulative probability function

Explain why the cumulative probability function in eq. (1.9) takes only values between 0 and 1.

Exercise 1.4: Cumulative and quantile functions of exponential random variables

Verify the formulas of $C(r)$ and $C^{-1}(r)$ in example 1.13.

Exercise 1.5: Sum of random variables

Consider two independent random variables R_1 and R_2 with densities $p_1(r_1)$ and $p_2(r_2)$, respectively. Show that the density $p_3(r_3)$ of a random variable R_3, with values $r_3 = r_1 + r_2$, is equal to the convolution $p_3(r_3) = (p_1 * p_2)(r_3)$.

Exercise 1.6: Minimum of exponential random variables

Consider two exponential random variables $R_1 \sim$ Exponential(λ_1) and $R_2 \sim$ Exponential(λ_2). Show that the random variable R_3, with values $r_3 = \min(r_1, r_2)$, follows an Exponential$(\lambda_1 + \lambda_2)$ distribution.

Exercise 1.7: A sanity check on random variable rescaling

Verify that the density $q(v)$ of the rescaled random variable V in example 1.7 has the correct units and that it is properly normalized.

Exercise 1.8: Linear transformations

In examples 1.7 and 1.8 we have seen how to obtain the probably density of random variables under rescaling and rotation. However, these two separate operations can be combined in a single more general one. For instance, consider a bivariate random variable (X, Y) and suppose that (X', Y') is the random variable under a linear transformation

$$x' = Ax + By + C, \qquad y' = Dx + Ey + F,$$

where A, B, C, D, E, and F are constants. Find the probability density of (X', Y') in terms of $p(x, y)$ and to avoid degeneracies consider only the case with $AE - BD \neq 0$.

Exercise 1.9: Division of random variables

Consider a random variable V with values $v = 1/x$, where X is a scalar random variable with density $p(x)$. Compute the density $q(v)$ in terms of $p(x)$.

Exercise 1.10: Spherical coordinate transformations

Consider a tri-variate random variable (X, Y, Z) that models a position in the Cartesian space. Use a transformation of random variables to relate the probability density $p(x, y, z)$ with the probability density $q(r, \phi, \theta)$ of the same position in spherical coordinates (R, Φ, Θ).

Exercise 1.11: Gamma random variables and derivatives

Suppose $R_1 \sim \text{Gamma}(\alpha_1, \beta)$ and $R_2 \sim \text{Gamma}(\alpha_2, \beta)$ are independent random variables. Find the probability densities of the random variables V_1, V_2, V_3 with values

$$v_1 = r_1 + r_2, \qquad v_2 = \frac{r_1}{r_1 + r_2}, \qquad v_3 = \frac{r_1}{r_2}.$$

Exercise 1.12: Manipulating transformed densities

Consider iid random variables R_1, R_2, R_3 with a common density $p(r)$. Further, assume R_1, R_2, R_3 are random variables that can only be realized as real and positive scalars. Find, in terms of $p(r)$, the probability that the polynomial $r_1 x^2 + r_2 x + r_3$ has real roots.

Exercise 1.13: The Weibull distribution

Consider a continuous random variable $X \sim \mathbb{P}$ that takes real scalar values and whose probability density is

$$p(x) = \frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha-1} e^{-\left(\frac{x}{\beta}\right)^{\alpha}}, \qquad x > 0$$

for appropriate values α and β.

1. Describe an algorithm that uses the fundamental theorem of simulation to simulate the random variable X.
2. Implement your algorithm.
3. Use simulations to verify that your implementation produce variables with the correct statistics.

Exercise 1.14: A fair dice

Use the fundamental theorem of simulation to simulate a roll of a fair dice. Generate several rolls and verify that indeed the dice simulated is fair.

Exercise 1.15: Label invariance of the fundamental theorem of simulation

1. Apply the fundamental theorem of simulation to simulate draws from Categorical$_{\rho_1,\rho_2,\rho_3}(\pi_{\rho_1},\pi_{\rho_2},\pi_{\rho_3})$.
2. Verify that ρ_1,ρ_2,ρ_3 are realized with probabilities $\pi_{\rho_1},\pi_{\rho_2},\pi_{\rho_3}$, respectively.
3. Apply a relabeling of ρ_1,ρ_2,ρ_3 and verify that the fundamental theorem of simulation keeps yielding realizations with the correct probabilities.

Exercise 1.16: Exponential rate

1. Sample 100 exponential random variables (the rate, λ must be specified by hand). Use your generated data to construct histograms of the CDF and PDF (with reasonable bin sizes selected at will).
2. Use the mean of your data to estimate the rate, λ. Then estimate the mean by fitting the CDF as well as the PDF (using whichever preferred criterion, like minimizing a mean square difference between the fit and the data, to find the best λ). Which of the three methods for estimating λ seems to yield more accurate results? Why?

Exercise 1.17: A loaded dice

A dice is rolled 120 times yielding the results:

face	"1"	"2"	"3"	"4"	"5"	"6"
number of appearances	15	34	18	19	19	15

Reason, based on a likelihood approach, that the dice is loaded.

Chapter 2

Dynamical systems and Markov processes

By the end of this chapter, we will have presented
- *Model formulations with stochastic dynamics*
- *The Markov property*
- *Common examples of dynamical systems*

In this chapter we focus on dynamical systems. Our systems change stochastically over time. We will be using the formulations introduced here in subsequent chapters. For clarity, we present in detail specific, tractable, examples that give rise to the development of Markov processes.

Note 2.1: Random processes

We want to clarify, from the very beginning, a subtlety in the terminology adopted in the literature. In mathematics and statistics the term *process*, most often an abbreviation for *a random process*, has a *very particular and technical* meaning. It is used to designate a *collection* of random variables that is, most commonly, *infinite*.

Although intuitive in its abstraction, the term process might be misleading in the context of dynamical systems as it conflicts with the way the word process is used in everyday language. For example, in everyday language a process pre-assumes some degree of temporal arrangement and, as such, entails some sense of causality. However, a temporal arrangement or any structure at all, is completely absent in the technical sense.

In this chapter and the rest of the book we deal with processes in the second, less technical, capacity that might not necessarily coincide with the mathematical one. Following this convention, a process is nothing more than a physical phenomenon that evolves over time. Of course, as we will see shortly, such phenomena are naturally formulated mathematically with random variables and, for this reason, very often our formulations end up being random processes in the technical capacity as well.

2.1 Why do we care about stochastic dynamical models?

The systems we consider are *stochastic*. That is, they are influenced by various random events and their output is random too. As with any random quantity, the system's output is sampled from appropriate probability distributions. This is by contrast to deterministic systems where their output is certain and sampling is unnecessary.

Our main objective, for now, is to introduce such distributions and to do so we develop appropriate forward or generative models. However, before we get into the fine details, we ask: *Why do we care about stochastic models?*

One answer to this question is that stochastic models are genuinely interesting and important as they often exhibit behavior *different* from their deterministic counter-parts. For example, genes in a living cell may be either active or suppressed and stochastic models may capture the random toggling between both gene states. A deterministic model, such as one developed on mean-field or mass-action principles, can instead capture only an "average" behavior over time of a gene. This fails to provide insight into the fact that this gene is either active

or not at any given time instant. While receiving much attention, arguments such as these on the inadequacy of averages are only a small part of why we need stochastic models.

Another, perhaps stronger, reason for studying stochastic systems is because measurement noise adds uncertainty which invariably introduces stochasticity. Thus, stochastic models are required to quantify the uncertainty introduced not only by the random events affecting the dynamics of the system, if warranted, but also by the unavoidable measurement noise. As we will see shortly, stochastic models can be translated into probabilities of sequences, or otherwise interrelated events, that are the starting point for quantitative data-driven analyses that we address in subsequent chapters.

Note 2.2: Signal processing

The analysis of the output of a dynamical system is sometimes termed *signal processing*. From our perspective, a signal is any quantity measured over time. In practice, we encounter four types of signals:

DSDT	DSCT
discrete state-space measured in discrete time	discrete state-space measured in continuous time
CSDT	**CSCT**
continuous state-space measured in discrete time	continuous state-space measured in continuous time

While physical considerations dictate that time be treated continuously, it is often convenient to discretize time. This is done in an effort to model data collection, that we will discuss later, or approximate the continuous time dynamics of a continuously evolving system.

2.2 Forward models of dynamical systems

Most often, we use *state variables* to describe a dynamical system. The term *state* designates the property or properties of the system that we represent as they evolve in time. A convenient way think of the state is as a set of features specifying our system. The precise meaning of these features, of course, depends on the specifics of the system at hand. For example, when studying a cell culture, the state may simply be the total population of cells in the culture. Alternatively, when studying particle motion, such as a diffusion of a particle in solution, the state may be its position with respect to some frame of reference.

For any given system, all possible values attained by its state are termed the *state-space*. For example, for a cell culture, in which we keep track of cell population, the state-space consists of the non-negative integers. Similarly, for a diffusing particle, the state-space consists of every point in the volume available to this particle.

As our system evolves in time, the value of its state changes. In other words, a system's state moves across the state-space. Monitored over a period of time, successive state values form the system's *trajectory*; fig. 2.1. In general, state positions within the state-space may be revisited from time to time and so, in general, a trajectory may reattain the same value multiple times. In addition, because a system may visit only a portion of its entire state-space, some states within the state-space may be absent from or remain unexplored by a trajectory.

Note 2.3: Distinction between constitutive and passing states

When we describe a dynamical system, it is essential to distinguish between a state that is a element of the system's state-space and a state that is a element of its trajectory. For such subtle cases, we will refer to the former as *constitutive* and the latter as *passing* states. See fig. 2.1.

The distinction between these two is made clearer if we consider that two different trajectories that the system can follow may consist of the *same* constitutive states; however, they contain *different* passing states. As we

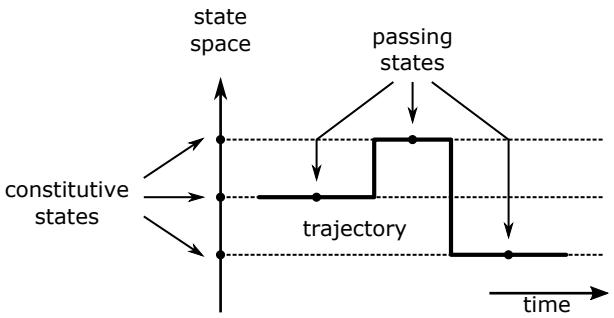


Figure 2.1: A generic description of a dynamical system. For clarity, on the vertical axis, the state-space is shown as having one dimension; however, a state-space may more generally have any number of dimensions.

will see shortly, mathematically we represent the passing states as random variables and the constitutive states as specific values that these random variables may attain.

Example 2.1: A cell culture as a dynamical system

Suppose that we are interested in studying how a population of cells in a culture changes over time and that we assess the culture at regular time intervals. In this setting, we consider the cell number measured at each time point as our system's state. Therefore, the state-space consists of the non-negative integers $0, 1, 2, \ldots$ or, in other words, the constitutive states are $0, 1, 2, \ldots$.

Further, suppose that s_n denotes the population measured at the n^{th} assessment. In this case, our system's trajectory is $(s_1, s_2, \ldots)$ and, following our convention, each of $s_1, s_2, \ldots$ is a passing state.

Different cell cultures, for example grown in separate Petri dishes, have the *same* constitutive states; however, since cell division happens at random times, different cell cultures generally follow *different* trajectories. Accordingly, although the constitutive states are the same for all cultures, the passing states at any given time may differ from culture to culture.

In subsequent chapters we will see how to study systems where the state is only *indirectly* observed. For example, the population s_n of a cell culture is often assessed by measuring the surface area of the Petri dish covered by the cells. However, because cells may grow over one another or because of the variability in the surface area covered by a single cell, the assessment of the culture's population s_n carried out this way is generally inaccurate. In such cases, we may speak of s_n as being a *hidden* state and we will learn how to formulate and analyze models with hidden passing states.

Depending on the context and the nature of the state-space, a system's states may attain *discrete* or *continuous* values as they evolve over time. For example, the population of cells in a growing culture is discrete while the position of a diffusing particle is continuous.

Despite the values a system's states may attain, all natural systems, including most physical and chemical ones, evolve *continuously in time*. This is certainly true of a mechanical system whose state (*i.e.* positions and momenta) evolves according to Newton's equations and equally true of a quantum system whose state (*i.e.* wavefunction) evolves according to the Schrödinger's equation. Nevertheless, as an idealization to continuous time evolution, we sometimes model systems and their states as evolving *discretely in time* as well.

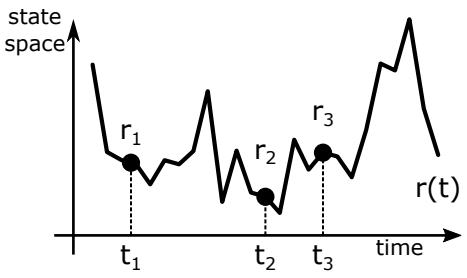


Figure 2.2: Although the state $\boldsymbol{r}(t)$ of a system may change continuously over time, we may model the system only at discrete times $t_1, t_2, t_3 \ldots$.

Example 2.2: Discrete and continuous time evolution

The position $\boldsymbol{r}(t) = (x(t), y(t), z(t))$ of a diffusing particle in solution is well defined for every time instance t. Thus, we say that the state of the system, *i.e.* the particle's position, evolves continuously; see fig. 2.2. However, positions may be assessed only at discrete time points $t_1, t_2, \ldots$. For simplicity, it may therefore be better to consider a discrete time evolution where we only keep track of $\boldsymbol{r}(t_1), \boldsymbol{r}(t_2), \ldots$ which, for simplicity, we may denote $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots$. In the latter setting, the position at times *between* the time points $t_1, t_2, \ldots$ may remain undefined. Generally, idealizations like this that lead to time-discretization are preferable as they lead to formulations that are mathematically more convenient.

The dynamical properties of a system depend on whether the system has a discrete or continuous state-space as well as whether it evolves in discrete or continuous time. All four possibilities: (i) systems with discrete state-spaces evolving in continuous time; (ii) systems with discrete state-spaces evolving in discrete time; (iii) systems with continuous state-spaces evolving in discrete time; and (iv) systems with continuous state-spaces evolving in continuous time, are of practical interest. Due to their unique characteristics, in the following sections we present each case separately.

Note 2.4: Description of a dynamical system

Irrespective of the details, a description of a dynamical system that is appropriate for quantitive analysis must specify:

- the state-space; *i.e. what are the system's states?*
- the initialization rule; *i.e. where does the system start?*
- the transition rules; *i.e. how does the system evolve?*

In subsequent chapters, where we will encounter systems with hidden states, we will have to include an additional feature:

- the assessment rules; *i.e. how is the system related to the measurements?*

2.3 Systems with discrete state-spaces in continuous time

By discrete system we mean a system whose constitutive states are separated from each other and they do *not* form a continuum. Since the system's state-space is discrete, we may universally denote the constitutive states with the label σ_m and use $m = 1, \ldots, M$, where M is the size of the state-space, to label them.

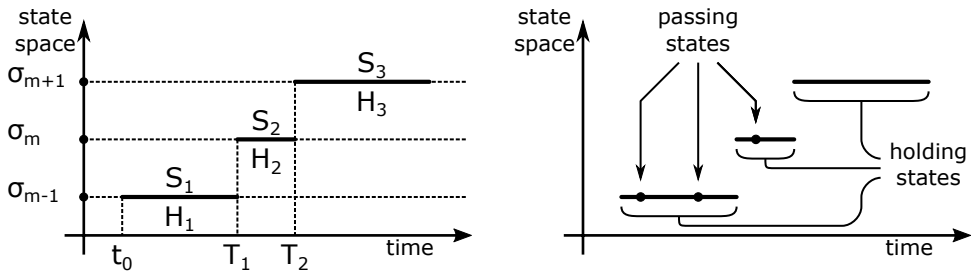


Figure 2.3: A system with a discrete state-space following continuous time dynamics. This system is described by holding states S_n and either jump times T_n or holding periods H_n.

Example 2.3: Examples of discrete systems

The analysis of experiments involving single molecule imaging often requires models of fluorescing molecules or *fluorophores* as they are commonly termed. In the simplest of cases, fluorophores can be modeled as attaining $M = 2$ states. Namely

- $\sigma_1 =$ light emitting (bright) state
- $\sigma_2 =$ light non-emitting (dark) state.

More detailed models may demand a greater number of constitutive states. For example, the state-space could be as follows

- $\sigma_1 =$ bright state
- $\sigma_2 =$ short lasting dark state
- $\sigma_3 =$ long lasting dark state
- $\sigma_4 =$ permanent dark state

where, now, $M = 4$. The latter case may more faithfully model real fluorophores as it accounts for multiple dark states and a permanent dark state that captures a phenomenon termed photobleaching.

As another example, large biological molecules, such as proteins, can attain multiple conformational states. For a protein undergoing transitions between folded and unfolded states, the state-space may be modeled as

- $\sigma_1 =$ folded
- $\sigma_2 =$ unfolded
- $\sigma_3 =$ partially folded.

Additional partially folded states may be further recruited as warranted by the biology.

Note 2.5: A labeling convention

The labels m used to distinguish the constitutive states, σ_m, of a discrete system are a mere convention and, generally, carry no particular meaning. In fact, instead of $\sigma_1, \sigma_2, \ldots$ we can very well chose a convention that does not rely on numerical labels. For example we could denote the constitutive states with $\alpha, \beta, \ldots$ or even $\spadesuit, \clubsuit, \ldots$ and still develop the same framework as described below.

We emphasize that we denote the states with $\sigma_1, \sigma_2, \ldots$ and we adopt *numerical* state labels $1, 2, \ldots$ for typographical reasons only. Unfortunately, in subtle situations, such numerical labels may be misleading as: (i) they may suggest that the underlying state-space has an ordering which, in general, is not needed; and (ii) shift the attention to the labels m rather than the states σ_m which are the primitive objects of interest.

The course of a system with a discrete state-space in continuous time, fig. 2.3, consists of *phases* (sometimes called *epochs* or *holding periods*) during which the system occupies the same constitutive states. Phases start and end precisely at the times at which the system switches from one constitutive state to another. For this reason, the system is fully described by the sequence of states attained in each phase $S_1, S_2, S_3, \ldots$ and the sequence of

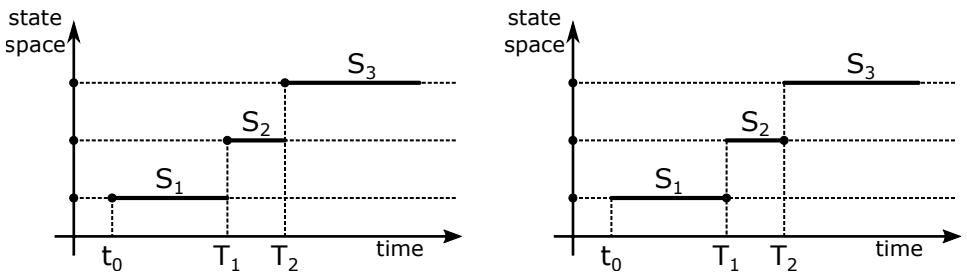


Figure 2.4: Examples of càllàl trajectories: the trajectory on the left is a càdlàg; while, the trajectory on the right is a càglàg.

times $T_1, T_2, \ldots$ at which the system jumps fro state to state. Specifically, we use T_n to denote the time at which the system jumps from S_n to S_{n+1} and, for this reason, call it the *jump time*. As $S_1, S_2, S_3, \ldots$ and $T_1, T_2, \ldots$ are understood as random variables, we can speak of the probability distributions from which they are sampled.

Our description is simplified if we adopt variables S_n, H_n rather than S_n, T_n as shown in fig. 2.3. In this description, S_n is the random variable representing the state of the system *before* the n^{th} jump and H_n is the random variable representing the period of time for which our system stays in S_n; see fig. 2.3. From now on, we will refer to H_n as a *holding period* and S_n as its coinciding *holding state*. As we can see from fig. 2.3, jump times and holding periods are related by

$$T_n = t_0 + H_1 + \cdots + H_n = t_0 + \sum_{n'=1}^{n} H_{n'} \tag{2.1}$$

where t_0 is a reference time that is, typically, not random. From eq. (2.1), we can easily deduce holding periods from jump times and *vice versa*.

Note 2.6: What is the trajectory?

A system's trajectory, with discrete state-space evolving in continuous time, is a function of time. We may denote this function with $\mathcal{S}(\cdot)$ and use functional notation to avoid any confusion that might be caused with the passing state $\mathcal{S}(t)$ at a particular time instant t.

According to fig. 2.3, the trajectory or its passing states are computed piecewise by

$$\mathcal{S}(t) = \begin{cases} S_1, & t_0 \le t < T_1 \\ S_2, & T_1 \le t < T_2 \\ \cdots \end{cases} \quad \text{or} \quad \mathcal{S}(t) = \begin{cases} S_1, & t_0 < t \le T_1 \\ S_2, & T_1 < t \le T_2 \\ \cdots \end{cases}.$$

Both of these are special cases of functions termed càllàl from a French acronym for *"continuous from one side with limit from the other"*. In particular, functions like $\mathcal{S}(\cdot)$ on the left are termed càdlàg for *"continuous on the right with limits on the left"* and functions like $\mathcal{S}(\cdot)$ on the right are termed *càglàg for "continuous on the left with limits on the right"*.

For the examples we will see in this chapter, which side (left or right) we place the equality on the jump times will not be so important. However, in subtle modeling cases, where multiple dynamical processes evolve simultaneously, such choice may have important consequences. When necessary, to emphasize the distinction in the subsequent chapters, at the jump times we often place filled dots on the appropriate state. For example, on fig. 2.4 we demonstrate this convention with the two trajectories given above.

We begin by describing the probability distributions over the states.

The initial holding state S_1, unaffected by the system's transitions at later times, is sampled separately from future holding states. As there are only discretely many choices for S_1, in general, initialization of the system is

described by

$$S_1 \sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\rho}) \tag{2.2}$$

where the probability array $\boldsymbol{\rho} = [\rho_{\sigma_1}, \ldots, \rho_{\sigma_M}]$ provides weights for particular initial states.

Causality suggests that sampling of a subsequent holding state may be affected only by preceding states. Under the *Markov assumption* that holding states only be affected by their immediate predecessor, the system's transitions are described by

$$S_{n+1}|s_n \sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\pi}_{s_n}) \tag{2.3}$$

where the probability array $\boldsymbol{\pi}_{s_n} = [\pi_{s_n \to \sigma_1}, \ldots, \pi_{s_n \to \sigma_M}]$ provides weights for particular transitions out of s_n. Since there are M constitutive states in the system's state-space, the transition rules entail M probability arrays $\boldsymbol{\pi}_{\sigma_1} = [\pi_{\sigma_1 \to \sigma_1}, \ldots, \pi_{\sigma_1 \to \sigma_M}], \ldots, \boldsymbol{\pi}_{\sigma_M} = [\pi_{\sigma_M \to \sigma_1}, \ldots, \pi_{\sigma_M \to \sigma_M}]$. As states must change at every jump time, these arrays exclude self-transitions, such that $\pi_{\sigma_1 \to \sigma_1} = \cdots = \pi_{\sigma_M \to \sigma_M} = 0$.

Together eqs. (2.2) and (2.3) entail $M+1$ probability arrays each consisting of M scalar values. Our dynamical model contains M initial weights for ρ_{σ_m} and a normalization constraint leading to $M-1$ free initial weights. Similarly, the remaining arrays have M^2 weights but also M normalization constraints and M weights fixed to zero; as such we have $M^2 - 2M$ free weights $\pi_{\sigma_m \to \sigma_{m'}}$.

Note 2.7: Transition probability matrix

Often it is convenient to tabulate the state-space and the transition probability arrays as

$$\begin{array}{c} \\ \sigma_1 \\ \vdots \\ \sigma_M \end{array} \begin{array}{c} \begin{array}{ccc} \sigma_1 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \pi_{\sigma_1 \to \sigma_1} & \cdots & \pi_{\sigma_1 \to \sigma_M} \\ \vdots & \ddots & \vdots \\ \pi_{\sigma_M \to \sigma_1} & \cdots & \pi_{\sigma_M \to \sigma_M} \end{bmatrix} \end{array} = \boldsymbol{\Pi}.$$

In this tabulation, all individual probability arrays out of one state arise as rows of $\boldsymbol{\Pi}$ which is commonly termed the *transition probability matrix*.

To avoid confusion with similar notions introduced later, it is useful to summarize some important characteristics of a transition probability matrix:

- It is a square matrix of size equal to the size of the state-space.
- Its row and column elements are arranged with the ordering of the state-space.
- It gathers unitless parameters.
- Each row must sum to 1, *i.e.* it is normalized to unit row sum.
- Its diagonal elements are 0.

Although sampling of the holding states $S_1, S_2, \ldots$ under the Markov assumption offers little modeling flexibility besides choosing the probability arrays $\boldsymbol{\rho}$ and $\boldsymbol{\pi}_{\sigma_m}$, sampling of the holding periods $H_1, H_2, \ldots$ is open to a variety of modeling choices and, as we will see below, different choices lead to different important examples. As each H_n attains positive scalar values, the only universal requirement for the sampling distributions is for them to exclusively allow positive values.

2.3.1 Renewal and Markov renewal processes

For dynamical systems known as *renewal processes*, each holding period H_n is independent of the others and sampled from the same distribution. In other words, the variables $H_1, H_2, \ldots$ are iid. The particular form for the sampling distribution differs from problem to problem.

Example 2.4: The birth process

With a birth process, we model the creation events of elements of some species $\mathcal{A}$.

In a birth process, the state-space is identified with the integers $0, 1, 2, \ldots$. The state s_n defines the population of species $\mathcal{A}$ just before the n^{th} birth event. In other words, the size of the state-space is infinite, $M = \infty$, and the constitutive states are $\sigma_m = m - 1$, for $m = 1, 2, \ldots$.

Typically, in a birth process we initially have k elements of $\mathcal{A}$. Subsequently, one element is added following each birth event. Under these conditions, the holding states read

$$s_n = k + n - 1, \qquad n = 1, 2, \ldots$$

Although the holding states are specified deterministically (and thus need not be sampled), we can also express them probabilistically. In particular, eqs. (2.2) and (2.3) for birth processes become

$$S_1 \sim \text{Categorical}_{0,1,2,\ldots}(\boldsymbol{\rho})$$
$$S_{n+1}|s_n \sim \text{Categorical}_{0,1,2,\ldots}(\boldsymbol{\pi}_{s_n})$$

where the transition probability arrays are given by the matrix

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & \cdots \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{array} = \boldsymbol{\Pi},$$

and initiating with $k = 0$ elements, the initial probability array is given by

$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & \cdots \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \end{bmatrix} = \boldsymbol{\rho}.$$

In a birth process, the holding periods are iid and often modeled as being sampled from an exponential distribution

$$H_n \sim \text{Exponential}(\lambda_b)$$

with a common birth rate λ_b.

Sampling each holding period using the same rate indicates that each element of $\mathcal{A}$ is produced by a source that is independent of the sources of the other elements. In other words, elements are created at a constant λ_b rate.

More generally, for dynamical systems known as *Markov renewal processes*, sampling of each holding period H_n may depend on its holding state s_n; unlike a renewal process where such dependence is not allowed. In this case, the holding periods $H_1, H_2, \ldots$ remain independent, similar to a renewal process; however, only holding periods associated with holding states in the same constitutive state are identically distributed.

Example 2.5: The death process

With a death process, we model the annihilation events of elements of some species $\mathcal{A}$.

In a death process, similar to a birth process, the state-space is identified with the integers $0, 1, 2, \ldots$. The state s_n designates the population of species $\mathcal{A}$ before the n^{th} death event.

As with birth processes, in a death process we initially have k elements of $\mathcal{A}$ with the understanding that one element is removed following each death event. Under these conditions, the holding states are specified

$$s_n = k - n + 1, \qquad n = 1, 2, \ldots, k.$$

As a population cannot go negative, the death process terminates at the k^{th} event. Just as with birth processes, there is no need to sample the state as the subsequent state is deterministic.

In a death process, the holding periods are independently sampled from an exponential distribution

$$H_n|s_n \sim \text{Exponential}\left(\lambda_{s_n}\right),$$

the rate of which depends on the holding states $\lambda_s = s\lambda_d$ where λ_d is the death rate.

Sampling each holding period using a rate proportional to the current population ($\lambda_s = s\lambda_d$) indicates that each element of $\mathcal{A}$ is annihilated due to a cause that is independent of the causes of the other elements. In other words, *each* element is annihilated at a constant rate which is λ_d.

This result is derived from the understanding that the rate of each holding period for s elements is obtained from the minimum of holding period for each of these elements treated separately; see appendix D.

Birth and death processes can be combined to model realistic scenarios involving simultaneous creation and annihilation events. We study this scenario, more generally, within the unified framework of Markov jump processes. These, as we will see shortly, are used to model systems with exclusively exponentially distributed holding periods. In any case, outside birth and death processes, a general renewal or Markov renewal process need not necessarily have exponential holding periods and, for this reason, they are often used to model systems with inherent memory.

2.3.2 Markov jump processes

Dynamical systems known as *Markov jump processes* are a specific kind of Markov renewal processes that are most commonly encountered when modeling physical systems. Due to their importance in the physical sciences, we discuss their properties in detail.

Modeling systems without memory

As Markov jump processes are examples of Markov renewal processes, each holding period H_n depends on its holding state s_n. In Markov jump process, however, holding periods are sampled from *memoryless* distributions. That is, the time period until the system jumps out of a passing state is independent of the time already spent in this state. As shown in appendix D, this means that the holding periods are sampled according to

$$H_n|s_n \sim \text{Exponential}\left(\lambda_{s_n}\right) \tag{2.4}$$

where rates may depend upon the holding states. In general, with M constitutive states, a Markov jump process entails M rates which are $\lambda_{\sigma_1}, \ldots, \lambda_{\sigma_M}$. Since these rates determine when a holding period ends, we call them *escape rates*.

Combining birth and death processes, already encountered in the context of renewal and Markov renewal processes, yields *birth-death processes* which are important examples of Markov jump processes.

Example 2.6: The birth-death process

With a birth-death process, we model the creation and annihilation events of elements of some species $\mathcal{A}$.

In a birth-death process, the state-space is identified by integers $0, 1, 2, \ldots$ and each state S_n with the population of species $\mathcal{A}$ before the n^{th} event, which now combines both births and deaths. In other words, the size of the state-space is infinite, $M = \infty$, and the constitutive states are $\sigma_m = m - 1$, for $m = 1, 2, \ldots$.

In a birth-death process we initially have k elements of species $\mathcal{A}$. Subsequently, one element is added or removed depending on whether a birth or death event was sampled, respectively. Unlike with pure birth or pure death processes, the holding states S_n cannot be expressed deterministically. Nevertheless, eqs. (2.2) and (2.3)

remain valid

$$S_1 \sim \text{Categorical}_{0,1,2,\dots}(\boldsymbol{\rho})$$
$$S_{n+1}|s_n \sim \text{Categorical}_{0,1,2,\dots}(\boldsymbol{\pi}_{s_n})$$

where the transition probability arrays are given by the matrix

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ \vdots \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & \cdots \end{array} \\ \begin{bmatrix} 0 & p_0 & 0 & 0 & \cdots \\ q_1 & 0 & p_1 & 0 & \cdots \\ 0 & q_2 & 0 & p_2 & \cdots \\ 0 & 0 & q_3 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{array} = \boldsymbol{\Pi}$$

with individual probabilities $p_s = \lambda_b/(s\lambda_d + \lambda_b)$ and $q_s = s\lambda_d/(s\lambda_d + \lambda_b)$, where λ_b and λ_d are the birth and death rates as discussed in appendix D.

In a birth-death process, the holding periods are independently sampled from an exponential distribution

$$H_n|s_n \sim \text{Exponential}(\lambda_{s_n}) \tag{2.5}$$

the rate of which depends on the holding states $\lambda_s = s\lambda_d + \lambda_b$.

Modeling elementary events

Physics typically defines elementary outcomes, by contrast to composite outcomes that reflect the net effect of multiple elementary outcomes, as having memoryless holding periods. The success of memoryless models explains why Markov jump process are very commonly found across physical applications.

As we have seen, to fully describe a Markov jump process on a state-space of size M, we need $M^2 - 2M$ scalar values to specify the transition probabilities that characterize the sequence of holding states $S_1, S_2, \dots$ and M rates that characterize the sequence of holding periods $H_1, H_2, \dots$. However, despite its mathematical convenience, a representation containing $\pi_{\sigma_m \to \sigma_{m'}}$ and λ_{σ_m} is not always the most physically interpretable. This is because, sampling of S_n and H_n are decoupled, suggesting that our system selects how long it stays in a state and the state to which it jumps, separately. However, from the physical point of view, these two events are better understood as a single event that can be described exclusively using the language of rates.

Fortunately, thanks to their lack of memory, Markov jump processes have an *equivalent* representation that more accurately reflects the selection event. In particular, since our system has M constitutive states, at any given time there are $M - 1$ constitutive states to which the system may jump. Accordingly, starting from a holding state s_n we can envision $M - 1$ concurrent *reactions* that compete to attract the system each to a different σ_m. Provided these reactions are memoryless, following appendix D, each one occurs after a period that is exponentially distributed with some rate that we denote $\lambda_{s_n \to \sigma_m}$.

From this viewpoint, the reaction that occurs *first* determines the next holding state s_{n+1} as well as the holding period h_n. Of course, consistency requires $\lambda_{s_n \to s_n} = 0$; but, otherwise the remaining $\lambda_{s_n \to \sigma_m}$ are free parameters.

In this representation, the transitions of a Markov jump process are fully specified by $M^2 - M$ reaction rates $\lambda_{\sigma_m \to \sigma_{m'}}$ that can be formed between any possible pair or constitutive states. At first, this agrees with the total number of transition weights $\pi_{\sigma_m \to \sigma_{m'}}$ *and* escape rates λ_{σ_m} needed in the earlier representation; however, as we spell out next, the connection is deeper.

We now re-express how to sample state transitions and holding times exclusively using rates with the understanding that we continue sampling S_1 as we did in eq. (2.2).

Now, as we understand s_{n+1} as the state that yields the minimum of the individual reaction periods, following note D.1, it is selected out of the constitutive states with a probability proportional to the corresponding rates

$\lambda_{s_n \to \sigma_m}$. Formally, this implies

$$S_{n+1}|s_n \sim \text{Categorical}_{\sigma_{1:M}}\left(\frac{\lambda_{s_n\to\sigma_1}}{\sum_{\sigma_m}\lambda_{s_n\to\sigma_m}}, \cdots, \frac{\lambda_{s_n\to\sigma_M}}{\sum_{\sigma_m}\lambda_{s_n\to\sigma_m}}\right). \tag{2.6}$$

Further, because the holding period h_n is the minimum of exponential periods, according to note D.1, it follows that it is exponentially distributed itself with a rate equal to the sum of the individual reaction rates

$$H_n|s_n \sim \text{Exponential}\left(\sum_{\sigma_m}\lambda_{s_n\to\sigma_m}\right). \tag{2.7}$$

Comparing eq. (2.6) with eq. (2.3) and eq. (2.7) with eq. (2.4), we immediately see that, not only are S_n and H_n sampled from similar distributions, but the parameters of these distributions are interrelated

$$\pi_{\sigma_m\to\sigma_{m'}} = \frac{\lambda_{\sigma_m\to\sigma_{m'}}}{\sum_{\sigma_{m'}}\lambda_{\sigma_m\to\sigma_{m'}}}, \qquad \lambda_{\sigma_m} = \sum_{\sigma_{m'}}\lambda_{\sigma_m\to\sigma_{m'}}.$$

As such, we can readily convert from one representation of a Markov jump process to the other at will.

Note 2.8: Transition rate matrix

Often it is convenient to tabulate the state-space and the reaction rates as

$$\begin{array}{c} \\ \sigma_1 \\ \vdots \\ \sigma_M \end{array}\begin{array}{c} \begin{array}{ccc} \sigma_1 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \lambda_{\sigma_1\to\sigma_1} & \cdots & \lambda_{\sigma_1\to\sigma_M} \\ \vdots & \ddots & \vdots \\ \lambda_{\sigma_M\to\sigma_1} & \cdots & \lambda_{\sigma_M\to\sigma_M} \end{bmatrix} \end{array} = \mathbf{\Lambda}.$$

In this tabulation, the reaction rates $\lambda_{\sigma_m\to\sigma_{m'}}$, which are non-negative scalars, are placed in the off diagonals. Further, self-reaction rates $\lambda_{\sigma_m\to\sigma_m}$, which are by definition zero, are placed along the diagonal. Commonly, $\mathbf{\Lambda}$ is termed the *transition rate matrix*.

To avoid confusion with similar notions that we will introduce later on, we summarize important characteristics of the transition rate matrix:

- It is a square matrix of size equal to the size of the state-space.
- Its row and column elements are arranged with the ordering of the state-space.
- It gathers parameters with units of reciprocal time (*i.e.* frequency).
- Rows need *not* be normalized.
- Its diagonal elements are zero.

Example 2.7: The switching process

With a *switching processes*, we model a system that cycles between a fixed number of constitutive states $\sigma_{1:M}$.

For concreteness, here we set $M = 3$. In this setting, the transition rate matrix is

$$\mathbf{\Lambda} = \begin{bmatrix} 0 & \lambda_{\sigma_1\to\sigma_2} & \lambda_{\sigma_1\to\sigma_3} \\ \lambda_{\sigma_2\to\sigma_1} & 0 & \lambda_{\sigma_2\to\sigma_3} \\ \lambda_{\sigma_3\to\sigma_1} & \lambda_{\sigma_3\to\sigma_2} & 0 \end{bmatrix}.$$

Given Λ, we can readily determine the the escape rates

$$\lambda_{\sigma_1} = \lambda_{\sigma_1\to\sigma_2} + \lambda_{\sigma_1\to\sigma_3}, \qquad \lambda_{\sigma_2} = \lambda_{\sigma_2\to\sigma_1} + \lambda_{\sigma_2\to\sigma_3}, \qquad \lambda_{\sigma_3} = \lambda_{\sigma_3\to\sigma_1} + \lambda_{\sigma_3\to\sigma_2}$$

as well as the transition probability matrix

$$\mathbf{\Pi} = \begin{bmatrix} \pi_{\sigma_1\to\sigma_1} & \pi_{\sigma_1\to\sigma_2} & \pi_{\sigma_1\to\sigma_3} \\ \pi_{\sigma_2\to\sigma_1} & \pi_{\sigma_2\to\sigma_2} & \pi_{\sigma_2\to\sigma_3} \\ \pi_{\sigma_3\to\sigma_1} & \pi_{\sigma_3\to\sigma_2} & \pi_{\sigma_3\to\sigma_3} \end{bmatrix} = \begin{bmatrix} 0 & \lambda_{\sigma_1\to\sigma_2}/\lambda_{\sigma_1} & \lambda_{\sigma_1\to\sigma_3}/\lambda_{\sigma_1} \\ \lambda_{\sigma_2\to\sigma_1}/\lambda_{\sigma_2} & 0 & \lambda_{\sigma_2\to\sigma_3}/\lambda_{\sigma_2} \\ \lambda_{\sigma_3\to\sigma_1}/\lambda_{\sigma_3} & \lambda_{\sigma_3\to\sigma_2}/\lambda_{\sigma_3} & 0 \end{bmatrix}.$$

The master equation

So far, we have presented Markov jump processes and we have discussed how to sample their holding states $S_1, S_2, \ldots$ and holding periods $H_1, H_2, \ldots$.

In view of eq. (2.1) and note 2.6, these are effectively complete descriptions of the system's trajectory and could be useful in simulating trajectories or computing probabilities associated with given trajectories.

Example 2.8: Probability of a Markov jump process trajectory

According to note 2.6 a trajectory $\mathcal{S}(\cdot)$ of a Markov jump process is a function of time. Here, we assume that this function has been observed between t_0 and some later time that we denote $t_{\max}$. Observing $\mathcal{S}(\cdot)$ means that we have available three pieces of information: (i) how many holding states the system occupied during the observation window; (ii) the precise sequence of these states; and (iii) the precise sequence of jump times. For clarity, we denote the number of states with N, the sequence of states undergone with $s_1, \ldots, s_N$, and the sequence of jump times with $t_1, \ldots, t_{N-1}$. Through eq. (2.1) we can recover the values of the first $N-1$ holding periods

$$h_1 = t_1 - t_0, \qquad h_2 = t_2 - t_1, \qquad \cdots \qquad h_{N-1} = t_{N-1} - t_{N-2},$$

while, for the last holding period, we can recover only a lower threshold $h_N > t_{\max} - t_{N-1}$. Therefore, according to the sampling scheme of eqs. (2.2) to (2.4), the probability of sampling the observed trajectory $\mathcal{S}(\cdot)$ is *proportional* to the product

$$\overbrace{\rho_{s_1}\lambda_{s_1}e^{-\lambda_{s_1}(t_1-t_0)}}^{\text{sample } s_1 \text{ and } h_1} \cdot \overbrace{\pi_{s_1\to s_2}\lambda_{s_2}e^{-\lambda_{s_2}(t_2-t_1)}}^{\text{sample } s_2 \text{ and } h_2} \cdots$$
$$\cdots \underbrace{\pi_{s_{N-2}\to s_{N-1}}\lambda_{s_{N-1}}e^{-\lambda_{s_{N-1}}(t_{N-1}-t_{N-2})}}_{\text{sample } s_{N-1} \text{ and } h_{N-1}} \cdot \underbrace{\pi_{s_{N-1}\to s_N}\left(1 - e^{-\lambda_{s_N}(t_{\max}-t_{N-1})}\right)}_{\text{sample } s_N \text{ and } h_N}.$$

The last term in brackets is obtained from the probability of not transitioning between $t_{\max}$ and t_{N-1}.

We distinguish this probability over observed trajectory from a probability over observing a specific sequence of holding states and holding periods from t_0 to t_N. This is because the probability over the latter is normalized over all holding states and holding periods for a fixed number of jump events while the former must also be normalized over all jump events that may have occurred.

In practice, we may not always be interested in the probability of a system's trajectory as integrating over all jump events rarely feasible. Rather, we can write down a probability over observing a specific sequence of holding states and holding periods conditioned on a known number of jump events.

Alternatively, we may also want to write down the probability of occupying a particular constitutive state for a given sequence of times. This naturally involves integrating over all jump events that may have occurred between these times. To write down this probability, it is convenient to derive an appropriate set of differential equations satisfied by this probability. As we will see, these differential equations are termed master equations.

To be more precise, we consider a Markov jump process with rate matrix $\mathbf{\Lambda}$ on a state-space $\sigma_{1:M}$. As usual, we denote the trajectory with $\mathcal{S}(\cdot)$ to emphasize that the trajectory is a function of time. Since $\mathcal{S}(\cdot)$ is random, the passing state evaluated at any time t, $\mathcal{S}(t)$, is also random. Following our convention, the values $\mathcal{S}(t)$ may attain are precisely the constitutive states σ_m. To compute the probabilities of passing from each σ_m, we wish

to write down a master equation satisfied by $P_{\sigma_m}(t)$ informally defined by

$$P_{\sigma_m}(t) = \text{Probability of } \left(\mathcal{S}(t) = \sigma_m\right).$$

To begin writing down the master equation, we consider the outcomes that contribute to the probability difference $P_{\sigma_m}(t+dt) - P_{\sigma_m}(t)$ provided dt is small enough: • the system starts at σ_m and jumps to some other $\sigma_{m'}$ (this negatively contributes to $P_{\sigma_m}(t+dt) - P_{\sigma_m}(t)$); or • the system starts at any other $\sigma_{m'}$ and jumps to σ_m (this positively contributes to $P_{\sigma_m}(t+dt) - P_{\sigma_m}(t)$).

Since the holding periods are memoryless, we can readily compute the probabilities of these two outcomes. In particular, by considering holding periods that start at t, following eq. (2.4), the probabilities are

$$P_{\sigma_m}(t) \int_0^{dt} dh\, \text{Exponential}\left(h; \lambda_{\sigma_m}\right) = P_{\sigma_m}(t)\left(1 - e^{-\lambda_{\sigma_m} dt}\right),$$
$$\sum_{m' \neq m} P_{\sigma_{m'}}(t) \int_0^{dt} dh\, \text{Exponential}\left(h; \lambda_{\sigma_{m'} \to \sigma_m}\right) = \sum_{m' \neq m} P_{\sigma_{m'}}(t)\left(1 - e^{-\lambda_{\sigma_{m'} \to \sigma_m} dt}\right).$$

Combining them, we see that

$$P_{\sigma_m}(t+dt) - P_{\sigma_m}(t) \approx P_{\sigma_m}(t)\left(e^{-\lambda_{\sigma_m} dt} - 1\right) + \sum_{m' \neq m} P_{\sigma_{m'}}(t)\left(1 - e^{-\lambda_{\sigma_{m'} \to \sigma_m} dt}\right) \tag{2.8}$$

with the approximation valid provided dt is sufficiently small such that no more than one jump can occur between t and $t+dt$. Dividing both sides by dt and taking the limit $dt \to 0^+$ yields an exact equality resulting in a differential equation

$$\frac{d}{dt} P_{\sigma_m}(t) = -\lambda_{\sigma_m} P_{\sigma_m}(t) + \sum_{m' \neq m} \lambda_{\sigma_{m'} \to \sigma_m} P_{\sigma_{m'}}(t). \tag{2.9}$$

Gathering the probabilities of all constitutive states into a probability array

$$\boldsymbol{P}(t) = \left[P_{\sigma_1}(t), \ldots, P_{\sigma_M}(t)\right], \tag{2.10}$$

the differential equation in eq. (2.9) takes a particularly simple form

$$\frac{d}{dt}\boldsymbol{P}(t) = \boldsymbol{P}(t)\boldsymbol{G}, \tag{2.11}$$

known as the *master equation*.

Note 2.9: Generator matrix

Often it is convenient to tabulate the state-space and all rates as

$$\begin{array}{c} \\ \sigma_1 \\ \vdots \\ \sigma_M \end{array} \begin{array}{c} \begin{array}{ccc} \sigma_1 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} -\lambda_{\sigma_1} & \cdots & \lambda_{\sigma_1 \to \sigma_M} \\ \vdots & \ddots & \vdots \\ \lambda_{\sigma_M \to \sigma_1} & \cdots & -\lambda_{\sigma_M} \end{bmatrix} \end{array} = \boldsymbol{G}.$$

In this tabulation the reaction rates $\lambda_{\sigma_m \to \sigma_{m'}}$, which are non-negative scalars, are placed in the off diagonals. By contrast, *negative* escape rates λ_{σ_m}, which are non-positive scalars, are placed along the diagonal. Commonly, $\boldsymbol{G}$ is termed the *generator matrix*. The generator $\boldsymbol{G}$ and the transition rate $\boldsymbol{\Lambda}$ matrices, that we have already seen, are tightly related and we can interconvert from one to the other.

To avoid confusion with similar notions, we summarize some important characteristics of the generator matrix:

- It is a square matrix of size equal to the size of the state-space.
- Its row and column elements are arranged with the ordering of the state-space.
- It gathers parameters with units of reciprocal time (*i.e.* frequency).
- Each row must sum to 0, *i.e.* it is normalized to zero row sum.
- Its diagonal elements may not be zero (except for the uninteresting case where all row elements are also zero).

At the initial time t_0, our trajectory passes from the first holding state, *i.e.* $\mathcal{S}(t_0) = S_1$, which is sampled according to eq. (2.2). As such, we immediately get $P_{\sigma_m}(t_0) = \rho_{\sigma_m}$, which provides the initial condition

$$\boldsymbol{P}(t_0) = \boldsymbol{\rho}. \tag{2.12}$$

Under this initial condition, we can integrate the master equation in time. Using the matrix exponential, the solution attains a general form

$$\boldsymbol{P}(t) = \boldsymbol{\rho}\,\mathbf{exp}\left((t - t_0)\boldsymbol{G}\right) \tag{2.13}$$

where we understand the meaning of the exponential of the matrix as defined in the box below.

Note 2.10: The matrix exponential

The exponential $\mathbf{exp}\left(\ell\boldsymbol{L}\right)$, where ℓ is a scalar and $\boldsymbol{L}$ is a square matrix, is defined by a sum

$$\mathbf{exp}(\ell\boldsymbol{L}) = \boldsymbol{I} + \ell\boldsymbol{L} + \frac{\ell^2}{2!}\boldsymbol{L}\boldsymbol{L} + \frac{\ell^3}{3!}\boldsymbol{L}\boldsymbol{L}\boldsymbol{L} + \cdots = \sum_{j=0}^{\infty}\frac{\ell^j}{j!}\boldsymbol{L}^j$$

where $\boldsymbol{I}$ is the identity matrix of size equal to the size of $\boldsymbol{L}$.

The m^{th} element of $\boldsymbol{P}(t)$ is now understood as the probability of starting in state dictated by $\boldsymbol{\rho}$ and, after some elapsed time t, landing in state σ_m irrespective of what trajectory (*i.e.* summing over trajectories) the system took over this time.

Using the general solution to the master equation, we can now see that the state probabilities $\boldsymbol{P}(t')$ at some time t' are related to the state probabilities $\boldsymbol{P}(t)$ as some time $t < t'$ through the property of the matrix exponential in note 2.10

$$\boldsymbol{P}(t') = \boldsymbol{\rho}\,\mathbf{exp}\left((t' - t_0)\boldsymbol{G}\right) = \boldsymbol{\rho}\,\mathbf{exp}\left((t - t_0)\boldsymbol{G}\right)\,\mathbf{exp}\left((t' - t)\boldsymbol{G}\right) = \boldsymbol{P}(t)\,\mathbf{exp}\left((t' - t)\boldsymbol{G}\right).$$

For this reason, the matrix $\mathbf{exp}\left((t' - t)\boldsymbol{G}\right)$ is termed *propagator*; it propagates the solution at time t to the solution at time t'. From now on, we will denote the propagator from an earlier time t to a subsequent time t', as follows

$$\boldsymbol{Q}^{t\to t'} = \mathbf{exp}\left((t' - t)\boldsymbol{G}\right). \tag{2.14}$$

We understand the $(\sigma_m, \sigma_{m'})$ element of $\boldsymbol{Q}^{t\to t'}$, $Q^{t\to t'}_{\sigma_m,\sigma_{m'}}$, as the probability of starting from state σ_m at time t and ending in $\sigma_{m'}$ at time t'.

Note 2.11: A sanity check on the master equation

The array of probabilities $\boldsymbol{P}(t)$ is, by definition, unitless. By contrast, $\boldsymbol{G}$ has units of reciprocal time, similar to d/dt, as can be verified from eqs. (2.9) and (2.11).

As our system must occupy some constitutive state at all times, the vector elements of $\boldsymbol{P}(t)$ must be normalized to unity, *i.e* $\sum_m P_{\sigma_m}(t) = 1$, for all t. We can verify that the total probability does not change over time using

eq. (2.9). In particular, the rate of change of the total probability is

$$\begin{aligned}\frac{d}{dt}\sum_m P_{\sigma_m}(t) &= \sum_m \frac{d}{dt}P_{\sigma_m}(t) = \sum_m \left(-\lambda_{\sigma_m}P_{\sigma_m}(t) + \sum_{m'\neq m}\lambda_{\sigma_{m'}\to\sigma_m}P_{\sigma_{m'}}(t)\right)\\ &= -\sum_m \lambda_{\sigma_m}P_{\sigma_m}(t) + \sum_m\left(\sum_{m'\neq m}\lambda_{\sigma_{m'}\to\sigma_m}P_{\sigma_{m'}}(t)\right)\\ &= -\sum_m \lambda_{\sigma_m}P_{\sigma_m}(t) + \sum_{m'}\left(\sum_{m\neq m'}\lambda_{\sigma_{m'}\to\sigma_m}P_{\sigma_{m'}}(t)\right)\\ &= -\sum_m \lambda_{\sigma_m}P_{\sigma_m}(t) + \sum_{m'}\lambda_{\sigma_m}P_{\sigma_{m'}}(t)\\ &= 0.\end{aligned}$$

Therefore, we can verify that the total probability is equal to unity. In particular, we immediately see that

$$\sum_m P_{\sigma_m}(t) = \sum_m P_{\sigma_m}(t_0) = \sum_m \rho_{\sigma_m} = 1$$

where we invoked the initial condition of eq. (2.12) in the second equality above.

The master equation in matrix form, eq. (2.11), and the propagator $\boldsymbol{Q}^{t\to t'}$ are valid provided $\boldsymbol{P}(t)$ is a *row* array as in eq. (2.10) *and* the generator matrix $\boldsymbol{G}$ is a formed as in note 2.9. In a computational implementation, these give rise to row-matrix operations. If, instead of a row array, we consider $\boldsymbol{P}(t)$ as a column array, as it is customary in programming environments better suited for column-matrix operations, then the master equation in eq. (2.11), the propagator $\boldsymbol{Q}^{t\to t'}$, the form of the generator $\boldsymbol{G}$, and the general solution in eq. (2.10) must be altered appropriately.

Example 2.9: Probability of a visiting a sequence of states at given times

It follows from the definition of $Q^{t\to t'}_{\sigma_m,\sigma_{m'}}$ that the probability of starting from state s_1 at t_1 and visiting $s_2,\cdots,s_N$ at times $t_2,\cdots,t_N$ is proportional to

$$\rho_{s_1}Q^{t_1\to t_2}_{s_1,s_2}\cdots Q^{t_{N-1}\to t_N}_{s_{N-1},s_N}.$$

2.3.3 Structured Markov jump processes*

As we have seen already, Markov jump processes are used to model systems without memory and we have already presented the basic theory needed to analyze a simple memoryless system. Now, we consider the analysis of *composite* systems. Specifically, we consider systems formed by multiple subsystems, which we will call *elements*, each one of which its memoryless and so it follows its own Markov jump process. As we will see, the composite system is also modeled by a Markov jump process which, due to the additional structure inherited by its elements, exhibits properties that simple systems do not.

Composite Markov jump processes

To begin, we consider the case that our system is formed by J similar elements. In other words, each element follows a Markov jump process with common constitutive states and transition rates. That is, we model a *congruent* system whose elements have the same state-space which, from hereonin, we designate as *elementary*.

*This is an advanced topic and could be skipped on a first reading.

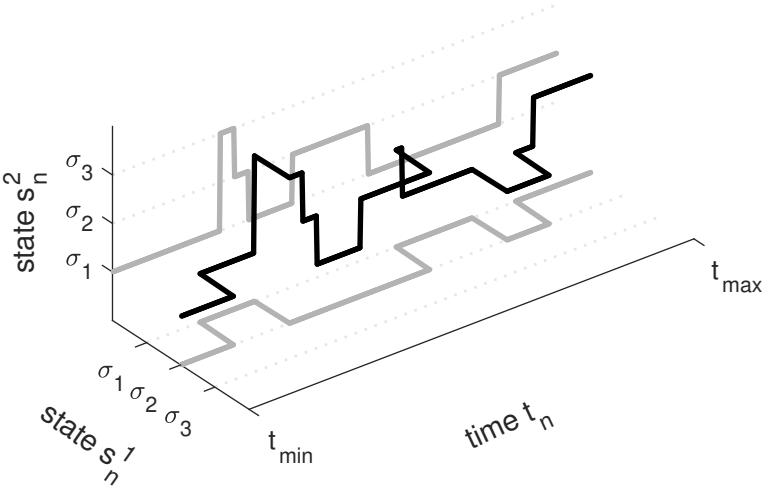


Figure 2.5: A composite system that moves over a state-space with two components.

When J elements are modeled together, they form a system whose holding states are best described by a J-tuple

$$\bar{\boldsymbol{s}}_n = \left(s_n^1, \ldots, s_n^j, \ldots, s_n^J\right)$$

where s_n^j is the passing state of the j^{th} element before the n^{th} jump event; see fig. 2.5. Of course, since in this description we consider the full system, n counts jump events from *all* elements.

For clarity, we will denote the composite state-space with $\bar{\boldsymbol{\sigma}}_{1:\bar{M}}$ and the elementary one with $\sigma_{1:M}$. Since each element has M constitutive states, the state-space of the composite system has $\bar{M} = M^J$ states.

Example 2.10: The composite state-space and rate matrix

For instance, if $M = 3$, the constitutive states of each element are $\sigma_1, \sigma_2, \sigma_3$ and the transition rate matrix of each element is

$$\boldsymbol{\Lambda} = \begin{bmatrix} 0 & \lambda_{\sigma_1\to\sigma_2} & \lambda_{\sigma_1\to\sigma_3} \\ \lambda_{\sigma_2\to\sigma_1} & 0 & \lambda_{\sigma_2\to\sigma_3} \\ \lambda_{\sigma_3\to\sigma_1} & \lambda_{\sigma_3\to\sigma_2} & 0 \end{bmatrix}.$$

For the special case of $J = 2$ elements, the state-space of the composite system has size $\bar{M} = 9$ and consists of

$$\begin{aligned} &\bar{\boldsymbol{\sigma}}_1 = (\sigma_1, \sigma_1), && \bar{\boldsymbol{\sigma}}_2 = (\sigma_1, \sigma_2), && \bar{\boldsymbol{\sigma}}_3 = (\sigma_1, \sigma_3), \\ &\bar{\boldsymbol{\sigma}}_4 = (\sigma_2, \sigma_1), && \bar{\boldsymbol{\sigma}}_5 = (\sigma_2, \sigma_2), && \bar{\boldsymbol{\sigma}}_6 = (\sigma_2, \sigma_3), \\ &\bar{\boldsymbol{\sigma}}_7 = (\sigma_3, \sigma_1), && \bar{\boldsymbol{\sigma}}_8 = (\sigma_3, \sigma_2), && \bar{\boldsymbol{\sigma}}_9 = (\sigma_3, \sigma_3). \end{aligned}$$

This is the *Cartesian product* $\bar{\boldsymbol{\sigma}}_{1:9} = \sigma_{1:3} \times \sigma_{1:3}$ of the elementary state-spaces. In a composite state-space as this one, it is customary to list the composite states $\bar{\boldsymbol{\sigma}}_m$ in *lexicographical* order.

With a careful examination of all possible transitions $\bar{\boldsymbol{\sigma}}_m \to \bar{\boldsymbol{\sigma}}_{m'}$, we can see that the composite system is

described by the rate matrix

$$
\begin{array}{c}
 \\
\bar{\boldsymbol{\sigma}}_1 \\
\bar{\boldsymbol{\sigma}}_2 \\
\bar{\boldsymbol{\sigma}}_3 \\
\bar{\boldsymbol{\sigma}}_4 \\
\bar{\boldsymbol{\sigma}}_5 \\
\bar{\boldsymbol{\sigma}}_6 \\
\bar{\boldsymbol{\sigma}}_7 \\
\bar{\boldsymbol{\sigma}}_8 \\
\bar{\boldsymbol{\sigma}}_9
\end{array}
\begin{array}{c}
\begin{array}{ccc|ccc|ccc}
\bar{\boldsymbol{\sigma}}_1 & \bar{\boldsymbol{\sigma}}_2 & \bar{\boldsymbol{\sigma}}_3 & \bar{\boldsymbol{\sigma}}_4 & \bar{\boldsymbol{\sigma}}_5 & \bar{\boldsymbol{\sigma}}_6 & \bar{\boldsymbol{\sigma}}_7 & \bar{\boldsymbol{\sigma}}_8 & \bar{\boldsymbol{\sigma}}_9
\end{array} \\
\left[\begin{array}{ccc|ccc|ccc}
0 & \lambda_{\sigma_1\to\sigma_2} & \lambda_{\sigma_1\to\sigma_3} & \lambda_{\sigma_1\to\sigma_2} & 0 & 0 & \lambda_{\sigma_1\to\sigma_3} & 0 & 0 \\
\lambda_{\sigma_2\to\sigma_1} & 0 & \lambda_{\sigma_2\to\sigma_3} & 0 & \lambda_{\sigma_1\to\sigma_2} & 0 & 0 & \lambda_{\sigma_1\to\sigma_3} & 0 \\
\lambda_{\sigma_3\to\sigma_1} & \lambda_{\sigma_3\to\sigma_2} & 0 & 0 & 0 & \lambda_{\sigma_1\to\sigma_2} & 0 & 0 & \lambda_{\sigma_1\to\sigma_3} \\
\hline
\lambda_{\sigma_2\to\sigma_1} & 0 & 0 & 0 & \lambda_{\sigma_1\to\sigma_2} & \lambda_{\sigma_1\to\sigma_3} & \lambda_{\sigma_2\to\sigma_3} & 0 & 0 \\
0 & \lambda_{\sigma_2\to\sigma_1} & 0 & \lambda_{\sigma_2\to\sigma_1} & 0 & \lambda_{\sigma_2\to\sigma_3} & 0 & \lambda_{\sigma_2\to\sigma_3} & 0 \\
0 & 0 & \lambda_{\sigma_2\to\sigma_1} & \lambda_{\sigma_3\to\sigma_1} & \lambda_{\sigma_3\to\sigma_2} & 0 & 0 & 0 & \lambda_{\sigma_2\to\sigma_3} \\
\hline
\lambda_{\sigma_3\to\sigma_1} & 0 & 0 & \lambda_{\sigma_3\to\sigma_2} & 0 & 0 & 0 & \lambda_{\sigma_1\to\sigma_2} & \lambda_{\sigma_1\to\sigma_3} \\
0 & \lambda_{\sigma_3\to\sigma_1} & 0 & 0 & \lambda_{\sigma_3\to\sigma_2} & 0 & \lambda_{\sigma_2\to\sigma_1} & 0 & \lambda_{\sigma_2\to\sigma_3} \\
0 & 0 & \lambda_{\sigma_3\to\sigma_1} & 0 & 0 & \lambda_{\sigma_3\to\sigma_2} & \lambda_{\sigma_3\to\sigma_1} & \lambda_{\sigma_3\to\sigma_2} & 0
\end{array}\right]
\end{array}
= \bar{\boldsymbol{\Lambda}}.
$$

In this matrix, besides self-transitions, all rates that correspond to double elementary jumps are 0, indicating that such transitions are excluded. In other words, by definition of a jump process, a composite system *cannot* undergo transitions for which more than one of its elements move across constitutive states.

The composite rate matrix has a particular block form that, informally, is expressed as

$$
\begin{array}{c}
 \\
\sigma_1 \times \sigma_{1:3} \\
\sigma_2 \times \sigma_{1:3} \\
\sigma_3 \times \sigma_{1:3}
\end{array}
\begin{array}{c}
\begin{array}{ccc}
\sigma_{1:3} \times \sigma_1 & \sigma_{1:3} \times \sigma_2 & \sigma_{1:3} \times \sigma_3
\end{array} \\
\begin{bmatrix}
\boldsymbol{\Lambda} & \lambda_{\sigma_1\to\sigma_2}\boldsymbol{I} & \lambda_{\sigma_1\to\sigma_3}\boldsymbol{I} \\
\lambda_{\sigma_2\to\sigma_1}\boldsymbol{I} & \boldsymbol{\Lambda} & \lambda_{\sigma_2\to\sigma_3}\boldsymbol{I} \\
\lambda_{\sigma_3\to\sigma_1}\boldsymbol{I} & \lambda_{\sigma_3\to\sigma_3}\boldsymbol{I} & \boldsymbol{\Lambda}
\end{bmatrix}
\end{array}
= \bar{\boldsymbol{\Lambda}}.
$$

Formally, such a matrix is obtained by

$$
\bar{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda} \otimes \boldsymbol{I}_M + \boldsymbol{I}_M \otimes \boldsymbol{\Lambda}
$$

where $\boldsymbol{I}_M$ is the identity matrix of size M and $\otimes$ is the matrix Kronecker product.

In general, every time the composite system leaves a holding state $\bar{\boldsymbol{s}}_n$, we may expect there to be $\bar{M}-1$ choices from which to select the subsequent holding state $\bar{\boldsymbol{S}}_{n+1}|\bar{\boldsymbol{s}}_n$. However, the majority of these choices involve more than one simultaneous elementary jump and, therefore, is prohibited. Consequently, leaving $\bar{\boldsymbol{s}}_n$, the composite system may choose one out of at most $J(M-1)$ composite constitutive states $\bar{\boldsymbol{\sigma}}_m$. These are precisely those $\bar{\boldsymbol{\sigma}}_m$ that involve only a single element's jump.

Example 2.11: Composite jump events

Continuing in the same setting as the previous example, if for some holding state $\bar{\boldsymbol{s}}_n = \bar{\boldsymbol{\sigma}}_1 = (\sigma_1, \sigma_1)$, then $\bar{\boldsymbol{S}}_{n+1}|\bar{\boldsymbol{s}}_n$ has 4 choices. Namely, these are

$$
\bar{\boldsymbol{\sigma}}_2 = (\sigma_1, \sigma_2), \qquad \bar{\boldsymbol{\sigma}}_3 = (\sigma_1, \sigma_3), \qquad \bar{\boldsymbol{\sigma}}_4 = (\sigma_2, \sigma_1), \qquad \bar{\boldsymbol{\sigma}}_7 = (\sigma_3, \sigma_1).
$$

The first two, $\bar{\boldsymbol{\sigma}}_2$ and $\bar{\boldsymbol{\sigma}}_3$, are triggered by jumps of the first element, $j = 1$; while, the remaining two, $\bar{\boldsymbol{\sigma}}_4$ and $\bar{\boldsymbol{\sigma}}_7$, are triggered by jumps of the second element, $j = 2$.

Similarly, if $\bar{\boldsymbol{s}}_n = \bar{\boldsymbol{\sigma}}_2 = (\sigma_1, \sigma_2)$, then $\bar{\boldsymbol{S}}_{n+1}|\bar{\boldsymbol{s}}_n$ has 4 different choices. Namely, now these are

$$
\bar{\boldsymbol{\sigma}}_1 = (\sigma_1, \sigma_1), \qquad \bar{\boldsymbol{\sigma}}_3 = (\sigma_1, \sigma_3), \qquad \bar{\boldsymbol{\sigma}}_5 = (\sigma_2, \sigma_2), \qquad \bar{\boldsymbol{\sigma}}_8 = (\sigma_3, \sigma_2).
$$

The first two, $\bar{\boldsymbol{\sigma}}_1$ and $\bar{\boldsymbol{\sigma}}_3$, are now triggered by jumps of the second element, $j = 2$; while, the remaining two, $\bar{\boldsymbol{\sigma}}_5$ and $\bar{\boldsymbol{\sigma}}_8$, are triggered by jumps of the first element, $j = 1$.

Each one of the J elements may trigger the transition $\bar{\boldsymbol{s}}_n \to \bar{\boldsymbol{s}}_{n+1}$. Therefore, a composite Markov jump process can be advanced by first sampling the identity J_n of the element triggering the n^{th} event and subsequently sampling the reaction that causes this element's jump. Therefore, in view of note D.1, a sampling scheme for composite systems proceeds as follows:

- First, sample J_n through

$$J_n|\bar{\boldsymbol{s}}_n \sim \text{Categorical}_{1:J}\left(\frac{\lambda_{s_n^1}}{\sum_j \lambda_{s_n^j}}, \cdots, \frac{\lambda_{s_n^J}}{\sum_j \lambda_{s_n^j}}\right). \tag{2.15}$$

- Then, sample $S_{n+1}^{j_n}$ through

$$S_{n+1}^{j_n}|j_n, \bar{\boldsymbol{s}}_n \sim \text{Categorical}_{\sigma_{1:M}}\left(\frac{\lambda_{s_n^{j_n}\to\sigma_1}}{\lambda_{s_n^{j_n}}}, \cdots, \frac{\lambda_{s_n^{j_n}\to\sigma_M}}{\lambda_{s_n^{j_n}}}\right).$$

- Finally, the remaining elements maintain their states

$$s_{n+1}^j = s_n^j, \qquad\qquad j \neq j_n.$$

This scheme is satisfactory for sampling systems consisting of few elements. However, for large systems, we can derive an alternative scheme that is more efficient. Before we describe this scheme, we first introduce some formalism.

With an elementary state-space of size M, there are at most $K = M^2 - M$ elementary reactions that are supported. Specifically, these are the reactions $\sigma_m \to \sigma_{m'}$ between any possible pair of different constitutive states. For clarity, we will label these reactions with $\gamma^{1:K}$. In view of note D.1, each one of them may cause the n^{th} jump with a rate equal to $c_{\bar{\boldsymbol{s}}_n}^{\sigma_m}\lambda_{\sigma_m\to\sigma_{m'}}$, where $c_{\bar{\boldsymbol{s}}_n}^{\sigma_m}$ is the total number of elements in $\bar{\boldsymbol{s}}_n$ occupying σ_m. We will denote the rate of a reaction γ^k with $\mu_{\bar{\boldsymbol{s}}_n}^k$ and, to distinguish it from the rates λ_{γ^k}, we will call it *propensity*. Similar to $c_{\bar{\boldsymbol{s}}_n}^{\sigma_m}$, we use subscripts to emphasize that propensities depend upon the holding state, $\bar{\boldsymbol{s}}_n$.

Example 2.12: Elementary reactions and propensities

In the setting of the previous examples, there are $K = 6$ elementary reactions that may be supported. Explicitly, these are

γ^1	γ^2	γ^3	γ^4	γ^5	γ^6
$\sigma_1 \to \sigma_2$	$\sigma_1 \to \sigma_3$	$\sigma_2 \to \sigma_1$	$\sigma_2 \to \sigma_3$	$\sigma_3 \to \sigma_1$	$\sigma_3 \to \sigma_2$.

The reactions triggered by $\bar{\boldsymbol{s}}_n = \bar{\boldsymbol{\sigma}}_1 = (\sigma_1, \sigma_1)$ and their respective propensities are

$\mu_{\bar{\boldsymbol{\sigma}}_1}^1$	$\mu_{\bar{\boldsymbol{\sigma}}_1}^2$	$\mu_{\bar{\boldsymbol{\sigma}}_1}^3$	$\mu_{\bar{\boldsymbol{\sigma}}_1}^4$	$\mu_{\bar{\boldsymbol{\sigma}}_1}^5$	$\mu_{\bar{\boldsymbol{\sigma}}_1}^6$
$2\cdot\lambda_{\sigma_1\to\sigma_2}$	$2\cdot\lambda_{\sigma_1\to\sigma_3}$	$0\cdot\lambda_{\sigma_2\to\sigma_1}$	$0\cdot\lambda_{\sigma_2\to\sigma_3}$	$0\cdot\lambda_{\sigma_3\to\sigma_1}$	$0\cdot\lambda_{\sigma_3\to\sigma_2}$

while, the reactions triggered by $\bar{\boldsymbol{s}}_n = \bar{\boldsymbol{\sigma}}_2 = (\sigma_1, \sigma_2)$ and their respective propensities are

$\mu_{\bar{\boldsymbol{\sigma}}_2}^1$	$\mu_{\bar{\boldsymbol{\sigma}}_2}^2$	$\mu_{\bar{\boldsymbol{\sigma}}_2}^3$	$\mu_{\bar{\boldsymbol{\sigma}}_2}^4$	$\mu_{\bar{\boldsymbol{\sigma}}_2}^5$	$\mu_{\bar{\boldsymbol{\sigma}}_2}^6$
$1\cdot\lambda_{\sigma_1\to\sigma_2}$	$1\cdot\lambda_{\sigma_1\to\sigma_3}$	$1\cdot\lambda_{\sigma_2\to\sigma_1}$	$1\cdot\lambda_{\sigma_2\to\sigma_3}$	$0\cdot\lambda_{\sigma_3\to\sigma_1}$	$0\cdot\lambda_{\sigma_3\to\sigma_2}$.

Under these definitions, an alternative scheme for advancing a composite Markov jump process proceeds by first sampling the reaction G_n causing the n^{th} event and subsequently sampling the element that triggers this reaction. In detail, the scheme is as follows:

- First, sample the reaction G_n that causes the transition through

$$G_n|\bar{\boldsymbol{s}}_n \sim \text{Categorical}_{\gamma^{1:K}}\left(\frac{\mu_{\bar{\boldsymbol{s}}_n}^1}{\sum_k \mu_{\bar{\boldsymbol{s}}_n}^k}, \ldots, \frac{\mu_{\bar{\boldsymbol{s}}_n}^K}{\sum_k \mu_{\bar{\boldsymbol{s}}_n}^k}\right). \tag{2.16}$$

- Then, sample the identify J_n of the element that causes this transition. The identity of the triggering element $J_n|g_n, \bar{\boldsymbol{s}}_n$ is now sampled among the elements in $\bar{\boldsymbol{s}}_n$ consistent with g_n. Since all of these are associated with the same escape rate λ_{g_n}, sampling of $J_n|g_n, \bar{\boldsymbol{s}}_n$ is uniform among the elements consistent with g_n.

- Finally, the remaining elements maintain their states

$$s^j_{n+1} = s^j_n, \qquad\qquad j \neq j_n.$$

In general, the total number, K, or supported reactions γ^k in a composite system cannot exceed $M^2 - M$; however, provided some elemental rates $\lambda_{\sigma_m \to \sigma_{m'}}$ are zero, K is lower. Although the second scheme is more complicated than the first one, in practice it can be executed faster. This is because, most often, the K of a composite system is considerably lower than the total number of elements, $K \ll J$. As we will see in the following sections, this situation is routinely encountered when modeling chemical systems and, for these systems, the second scheme is the only computationally feasible choice.

Note 2.12: Holding periods in a composite Markov jump process

In the *first scheme* the transition $\bar{\boldsymbol{s}}_n \to \bar{\boldsymbol{s}}_{n+1}$ is triggered by the element sampled in eq. (2.15), indicating that the holding period is determined by

$$H_n | \bar{\boldsymbol{s}}_n \sim \text{Exponential}\left(\sum_j \lambda_{s^j_n}\right).$$

Similarly, in the *second scheme*, the transition $\bar{\boldsymbol{s}}_n \to \bar{\boldsymbol{s}}_{n+1}$ is caused by the reaction sampled in eq. (2.16), indicating that the holding period is determined by

$$H_n | \bar{\boldsymbol{s}}_n \sim \text{Exponential}\left(\sum_k \mu^k_{\bar{\boldsymbol{s}}_n}\right).$$

Nevertheless, both exponential rates are equal

$$\sum_k \mu^k_{\bar{\boldsymbol{s}}_n} = \sum_m \sum_{m'} c^{\sigma_m}_{\bar{\boldsymbol{s}}_n} \lambda_{\sigma_m \to \sigma_{m'}} = \sum_m c^{\sigma_m}_{\bar{\boldsymbol{s}}_n} \sum_{m'} \lambda_{\sigma_m \to \sigma_{m'}} = \sum_m c^{\sigma_m}_{\bar{\boldsymbol{s}}_n} \lambda_{\sigma_m} = \sum_j \lambda_{s^j_n},$$

therefore, the holding periods are sampled similarly in both schemes.

Up to this point, we have focused on composite Markov jump processes formed by similar elements. However, this is not a necessary requirement. In fact, we may encounter composite systems whose elementary state-spaces differ. For example, the j^{th} element can have a state-space $\sigma^j_{1:M^j}$ of different size M^j than the others and may also have different elementary reaction rates $\lambda_{\sigma^j_m \to \sigma^j_{m'}}$. As the possibilities of combining elementary systems are endless, below we give a specific example open to generalizations.

Example 2.13: Incongruent composite systems

In this example, we consider an incongruent system formed by $J = 2$ elements where the first element has $M^1 = 2$ constitutive states that we denote σ^1_1, σ^1_2 and the second element has $M^2 = 3$ constitutive states that we denote $\sigma^2_1, \sigma^2_2, \sigma^2_3$.

As the two elements differ, their reactions are described by different rate matrices

$$\boldsymbol{\Lambda}^1 = \begin{bmatrix} 0 & \lambda_{\sigma^1_1 \to \sigma^1_2} \\ \lambda_{\sigma^1_2 \to \sigma^1_1} & 0 \end{bmatrix}, \qquad \boldsymbol{\Lambda}^2 = \begin{bmatrix} 0 & \lambda_{\sigma^2_1 \to \sigma^2_2} & \lambda_{\sigma^2_1 \to \sigma^2_3} \\ \lambda_{\sigma^2_2 \to \sigma^2_1} & 0 & \lambda_{\sigma^2_2 \to \sigma^2_3} \\ \lambda_{\sigma^2_3 \to \sigma^2_1} & \lambda_{\sigma^2_3 \to \sigma^2_2} & 0 \end{bmatrix}.$$

The composite state-space contains $M^1 M^2 = 6$ constitutive states resulting from the Cartesian product of the elementary state-spaces $\bar{\boldsymbol{\sigma}}_{1:6} = \sigma^1_{1:2} \times \sigma^2_{1:3}$. In particular

$$\bar{\boldsymbol{\sigma}}_1 = (\sigma^1_1, \sigma^2_1), \qquad \bar{\boldsymbol{\sigma}}_2 = (\sigma^1_1, \sigma^2_2), \qquad \bar{\boldsymbol{\sigma}}_3 = (\sigma^1_1, \sigma^2_3),$$
$$\bar{\boldsymbol{\sigma}}_4 = (\sigma^1_2, \sigma^2_1), \qquad \bar{\boldsymbol{\sigma}}_5 = (\sigma^1_2, \sigma^2_2), \qquad \bar{\boldsymbol{\sigma}}_6 = (\sigma^1_2, \sigma^2_3).$$

The transition rate matrix of the composite system is

$$\begin{array}{c} \\ \bar{\boldsymbol{\sigma}}_1 \\ \bar{\boldsymbol{\sigma}}_2 \\ \bar{\boldsymbol{\sigma}}_3 \\ \bar{\boldsymbol{\sigma}}_4 \\ \bar{\boldsymbol{\sigma}}_5 \\ \bar{\boldsymbol{\sigma}}_6 \end{array} \begin{array}{c} \begin{array}{cccccc} \bar{\boldsymbol{\sigma}}_1 & \bar{\boldsymbol{\sigma}}_2 & \bar{\boldsymbol{\sigma}}_3 & \bar{\boldsymbol{\sigma}}_4 & \bar{\boldsymbol{\sigma}}_5 & \bar{\boldsymbol{\sigma}}_6 \end{array} \\ \left[\begin{array}{cccccc} 0 & \lambda_{\sigma_1^2\to\sigma_2^2} & \lambda_{\sigma_1^2\to\sigma_3^2} & \lambda_{\sigma_1^1\to\sigma_2^1} & 0 & 0 \\ \lambda_{\sigma_2^2\to\sigma_1^2} & 0 & \lambda_{\sigma_2^2\to\sigma_3^2} & 0 & \lambda_{\sigma_1^1\to\sigma_2^1} & 0 \\ \lambda_{\sigma_3^2\to\sigma_1^2} & \lambda_{\sigma_3^2\to\sigma_2^2} & 0 & 0 & 0 & \lambda_{\sigma_1^1\to\sigma_2^1} \\ \lambda_{\sigma_2^1\to\sigma_1^1} & 0 & 0 & 0 & \lambda_{\sigma_1^2\to\sigma_2^2} & \lambda_{\sigma_1^2\to\sigma_3^2} \\ 0 & \lambda_{\sigma_2^1\to\sigma_1^1} & 0 & \lambda_{\sigma_2^2\to\sigma_1^2} & 0 & \lambda_{\sigma_2^2\to\sigma_3^2} \\ 0 & 0 & \lambda_{\sigma_2^1\to\sigma_1^1} & \lambda_{\sigma_3^2\to\sigma_1^2} & \lambda_{\sigma_3^2\to\sigma_2^2} & 0 \end{array}\right] \end{array} = \bar{\boldsymbol{\Lambda}}.$$

As can be seen, the composite matrix maintains its block form that is still obtained through Kronecker products

$$\bar{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}^1 \otimes \boldsymbol{I}_{M^2} + \boldsymbol{I}_{M^1} \otimes \boldsymbol{\Lambda}^2.$$

The composite transition rate matrix can also result by a single operation

$$\bar{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}^1 \oplus \boldsymbol{\Lambda}^2$$

where $\oplus$ denotes the matrix Kronecker sum which, by definition, is equal to $\boldsymbol{\Lambda}^1 \oplus \boldsymbol{\Lambda}^2 = \boldsymbol{\Lambda}^1 \otimes \boldsymbol{I}_{M^2} + \boldsymbol{I}_{M^1} \otimes \boldsymbol{\Lambda}^2$.

As illustrated by this example, which uses the Cartesian products and Kronecker sums, we can readily obtain the state-space and rate matrix resulting from the combination of any number of elementary systems. For instance, in the most general case, combining J different state-spaces

$$\sigma^1_{1:M^1}, \ldots, \sigma^j_{1:M^j}, \ldots, \sigma^J_{1:M^J},$$

with rate matrices

$$\boldsymbol{\Lambda}^1, \ldots, \boldsymbol{\Lambda}^j, \cdots, \boldsymbol{\Lambda}^J,$$

respectively, results in a composite state-space of size

$$\bar{M} = M^1 \cdots M^j \cdots M^J.$$

The composite state-space is given by

$$\bar{\boldsymbol{\sigma}}_{1:\bar{M}} = \sigma^1_{1:M^1} \times \cdots \times \sigma^j_{1:M^j} \times \cdots \times \sigma^J_{1:M^J}$$

and the composite rate matrix is given by

$$\bar{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}^1 \oplus \cdots \oplus \boldsymbol{\Lambda}^j \oplus \cdots \oplus \boldsymbol{\Lambda}^K.$$

Collapsed Markov jump processes

We have considered elements of a composite system that do not interact. However, this assumption is quite restrictive. For example, chemical systems involving multi-molecular reactions must be modeled as interacting elements and cannot be accommodated within our existing framework.

In this section, we relax this assumption. To do so, we adopt a new formulation of composite systems that ignores the identities of the individual elements forming the system but rather focuses on the *total population* of elements occupying each elementary state. This is a bulk formulation and, whenever we can afford it, *i.e.* whenever we can relax keeping track of the precise identity of the elements, as we will see, allows for a particularly convenient way to study our system. Indeed, we have already seen a special case of this collapsed formulation in birth and death processes.

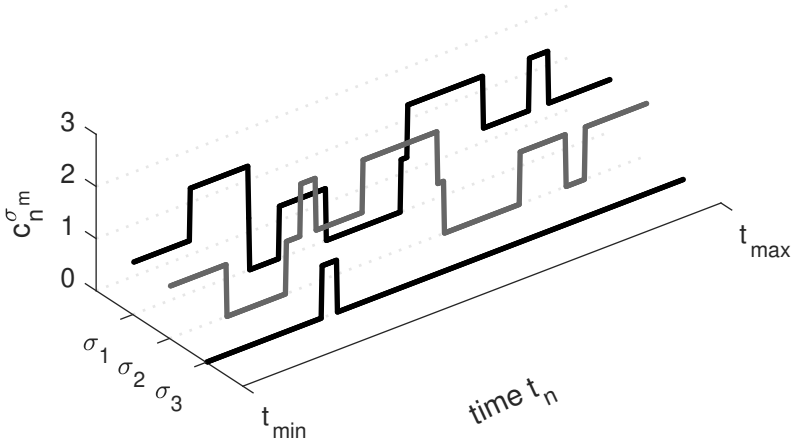


Figure 2.6: A collapsed description of a composite system.

As before, we consider J elements, each having M constitutive states $\sigma_{1:M}$. Further, we consider K elemental reactions $\sigma_m \to \sigma_{m'}$ labeled with $\gamma^{1:K}$. Now, instead of seeking a detailed description of the composite system through $\bar{\boldsymbol{s}}_n$, we keep track of a *collapsed* state. This is an M-tuple

$$\tilde{\boldsymbol{c}}_n = [c_n^{\sigma_1}, \ldots, c_n^{\sigma_m}, \ldots, c_n^{\sigma_M}]$$

where $c_n^{\sigma_m}$ is the population of elements occupying state σ_m before the n^{th} event; see fig. 2.6.

Note 2.13: The collapsed state-space

The state-space of a collapsed Markov jump process consists of all M-tuples $[c^{\sigma_1}, \cdots, c^{\sigma_m}, \cdots, c^{\sigma_M}]$, formed by the non-negative integers c^{σ_m} with $\sum_{\sigma_m} c^{\sigma_m} = J$. For example, for $M = 3$ and $J = 2$, the collapsed state-space consists of

$$[2,0,0], \qquad [0,2,0], \qquad [0,0,2], \qquad [1,1,0], \qquad [1,0,1], \qquad [0,1,1].$$

As it only keeps track of the total population in each state and not the states of all individual elements, a collapsed state-space is smaller than a composite state-space.

As we have seen in the preceding section, the propensities of the elementary reactions $\gamma^{1:K}$ depend only on the populations of elements in each σ_m, *i.e.* the collapsed state $\tilde{\boldsymbol{c}}_n$, and reaction rates. For clarity, from now on, we will denote the propensity of a reaction γ^k with $\mu^k_{\tilde{\boldsymbol{c}}_n} = c_n^{\sigma_m} \lambda_{\gamma^k}$, where λ_{γ^k} is the reaction rate of γ^k and, as earlier, we will use subscripts to emphasize that propensities depend upon $\tilde{\boldsymbol{c}}_n$.

In light of eq. (2.16), the reaction G_n triggering the transition $\tilde{\boldsymbol{c}}_n \to \tilde{\boldsymbol{c}}_{n+1}$ is sampled according to

$$G_n | \tilde{\boldsymbol{c}}_n \sim \text{Categorical}_{\gamma^{1:K}} \left(\frac{\mu^1_{\tilde{\boldsymbol{c}}_n}}{\sum_k \mu^k_{\tilde{\boldsymbol{c}}_n}}, \ldots, \frac{\mu^K_{\tilde{\boldsymbol{c}}_n}}{\sum_k \mu^k_{\tilde{\boldsymbol{c}}_n}} \right).$$

Once g_n is sampled, $\tilde{\boldsymbol{c}}_{n+1}$ is determined by updating the populations of the elements involved in g_n without sampling any additional random variables. According to note 2.12, the holding period H_n is then sampled from

$$H_n | \tilde{\boldsymbol{c}}_n \sim \text{Exponential} \left(\sum_k \mu^k_{\tilde{\boldsymbol{c}}_n} \right).$$

As we will see shortly, this sampling scheme constitute the foundation of the *Gillespie simulation*.

Note 2.14: Encoding elementary reactions

At this point we summarize some nomenclature regarding the formulation of elementary reactions.

A composite system, irrespective of whether we model it through detailed $\bar{\boldsymbol{S}}_1, \bar{\boldsymbol{S}}_2, \ldots$ or collapsed $\tilde{\boldsymbol{C}}_1, \tilde{\boldsymbol{C}}_2, \ldots$ holding states, entails an elementary state-space $\sigma_{1:M}$. This state-space may support at most $M^2 - M$ elementary reactions. We encode the elementary reactions with $\sigma_m \to \sigma_{m'}$, where σ_m and $\sigma_{m'}$ are two different elementary constitutive states and associate them with an elementary reaction rate $\lambda_{\sigma_m \to \sigma_{m'}}$. In $\sigma_m \to \sigma_{m'}$, we designate σ_m as the *departing* state and $\sigma_{m'}$ as the *arriving* state.

A system may have some elementary reaction rates equal to zero. Accordingly, we may say that the system cannot support these reactions. In such cases, which are very common in applications involving composite systems, the number of supported elementary reactions K is lower than $M^2 - M$. Occasionally, we denote the supported elementary reactions with $\gamma^{1:K}$, *i.e.* each γ^k stands for some $\sigma_m \to \sigma_{m'}$. Following this convention, we may also denote the elementary reaction rates using λ_{γ^k}, instead of the more elaborate $\lambda_{\sigma_m \to \sigma_{m'}}$, and we may further refer to the associated σ_m and $\sigma_{m'}$ as the reaction's departing and arriving states, respectively.

Besides a rate λ_{γ^k}, each γ^k is also associated with a propensity. Depending on the modeling description chosen, we denote the propensity of γ^k with either $\mu^k_{\bar{\mathbf{s}}_n}$ or $\mu^k_{\tilde{\mathbf{c}}_n}$. In either case, a reaction's propensity is equal to the product of elements occupying the departing state and the reaction's rate.

In a collapsed description of a composite system, it is convenient to associate each reaction γ^k with a *stoichiometric array* $\tilde{\boldsymbol{\zeta}}^k$ that indicates the population changes induced by a reaction. Similar to a collapsed state $\tilde{\mathbf{c}}$, each stoichiometric array is also an M-tuple

$$\tilde{\boldsymbol{\zeta}}^k = \left[\zeta^k_{\sigma_1}, \ldots, \zeta^k_{\sigma_m}, \ldots, \zeta^k_{\sigma_M}\right]$$

with each ζ^k_σ encoding how the population of an elementary constitutive state σ is affected by the reaction γ^k. For the reactions we have encountered so far, $\zeta^k_\sigma = -1$ for the departing and $\zeta^k_\sigma = +1$ for the arriving state of γ^k. In subsequent sections, we will encounter reactions with more complicated propensity. For this reason, it is most convenient to tabulate a system's reactions $\gamma^{1:K}$ and their stoichiometric arrays $\tilde{\boldsymbol{\zeta}}^{1:K}$ as

$$\begin{array}{c} \\ \gamma^1 \\ \vdots \\ \gamma^K \end{array} \begin{array}{c} \begin{array}{ccc} \sigma_1 & \cdots & \sigma_M \end{array} \\ \begin{pmatrix} \zeta^1_{\sigma_1} & \cdots & \zeta^1_{\sigma_M} \\ \vdots & \ddots & \vdots \\ \zeta^K_{\sigma_1} & \cdots & \zeta^K_{\sigma_M} \end{pmatrix} \end{array} = \begin{pmatrix} \tilde{\boldsymbol{\zeta}}^1 \\ \vdots \\ \tilde{\boldsymbol{\zeta}}^K \end{pmatrix}.$$

For example, a system with $M = 3$ elementary states, can support at most $K = 6$ elementary reactions. These are tabulated in

$$\begin{array}{c} \\ \gamma^1 \\ \gamma^2 \\ \gamma^3 \\ \gamma^4 \\ \gamma^5 \\ \gamma^6 \end{array} \begin{array}{c} \begin{array}{ccc} \sigma_1 & \sigma_2 & \sigma_3 \end{array} \\ \begin{pmatrix} -1 & +1 & 0 \\ -1 & 0 & +1 \\ +1 & -1 & 0 \\ 0 & -1 & +1 \\ +1 & 0 & -1 \\ 0 & +1 & -1 \end{pmatrix} \end{array} = \begin{pmatrix} \tilde{\boldsymbol{\zeta}}^1 \\ \tilde{\boldsymbol{\zeta}}^2 \\ \tilde{\boldsymbol{\zeta}}^3 \\ \tilde{\boldsymbol{\zeta}}^4 \\ \tilde{\boldsymbol{\zeta}}^5 \\ \tilde{\boldsymbol{\zeta}}^6 \end{pmatrix}.$$

The advantage of tabulating $\tilde{\boldsymbol{\zeta}}^{1:K}$ is that, following the sampling of a reaction G_n, the new occupying populations are readily obtained from $\tilde{\mathbf{c}}_{n+1} = \tilde{\mathbf{c}}_n + \tilde{\boldsymbol{\zeta}}^{g_n}$.

Master equations for composite and collapsed Markov jump processes

As the composite systems of the preceding section are Markov jump processes in their own right, appropriate master equations can also be used for their study. Here, we derive these equations. As the two descriptions we

have presented so far require different treatment, we present their derivation separately. As usual, we consider a composite system formed by J elements, each on a common elementary state-space $\sigma_{1:M}$ with elementary reaction rates gathered in $\boldsymbol{\Lambda}$.

Composite Markov jump process In a detailed description of a composite Markov jump process, the probabilities we seek are

$$\bar{P}_{\bar{\sigma}_m}(t) = \text{Probability of } \left(\bar{\boldsymbol{\mathcal{S}}}(t) = \bar{\boldsymbol{\sigma}}_m\right),$$

where now $\bar{\boldsymbol{\mathcal{S}}}(\cdot)$ denotes the composite trajectory which is a function of time and attains values in the composite state-space drawn from the Cartesian products $\bar{\boldsymbol{\sigma}}_{1:\bar{M}} = \sigma_{1:M} \times \cdots \times \sigma_{1:M}$. The composite rate matrix $\bar{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda} \oplus \cdots \oplus \boldsymbol{\Lambda}$ is formed by Kronecker sums and it is easy to verify that the same is true for the composite generator $\bar{\boldsymbol{G}} = \boldsymbol{G} \oplus \cdots \oplus \boldsymbol{G}$. Therefore, the master equation, eq. (2.13), for composite systems is

$$\frac{d}{dt}\bar{\boldsymbol{P}}(t) = \bar{\boldsymbol{P}}(t)\bar{\boldsymbol{G}} = \bar{\boldsymbol{P}}(t)\left(\boldsymbol{G} \oplus \cdots \oplus \boldsymbol{G}\right),$$

and, because $\oplus$ commutes with matrix exponentiation, its general solution is

$$\bar{\boldsymbol{P}}(t) = \bar{\boldsymbol{\rho}} \exp\left((t-t_0)\bar{\boldsymbol{G}}\right) = \bar{\boldsymbol{\rho}} \exp\left((t-t_0)\boldsymbol{\Lambda}^1\right) \otimes \cdots \otimes \exp\left((t-t_0)\boldsymbol{\Lambda}^J\right)$$

where $\bar{\boldsymbol{\rho}}$ denotes the initial probability vector of the composite system.

Collapsed Markov jump process Unlike composite Markov jump processes whose master equation we can derive by exclusively relying on properties of the matrix exponential, for collapsed Markov jump processes, we need to repeat the derivation from the onset. For this, we consider K reactions $\gamma^{1:K}$ with associated stoichiometric array $\tilde{\boldsymbol{\zeta}}^{1:K}$ as in note 2.14. The derivation parallels section 2.3.2 and, as such, we only highlight key steps.

In the collapsed formulation, the probabilities we seek are

$$\tilde{P}_{\tilde{\boldsymbol{c}}}(t) = \text{Probability of } \left(\tilde{\boldsymbol{\mathcal{S}}}(t) = \tilde{\boldsymbol{c}}\right),$$

where now $\tilde{\boldsymbol{\mathcal{S}}}(\cdot)$ denotes the collapsed trajectory. This is a function of time that takes values $\tilde{\boldsymbol{c}} = [c^{\sigma_1}, \ldots, c^{\sigma_M}]$. To derive the master equation for $\tilde{P}_{\tilde{\boldsymbol{c}}}(t)$ we consider changes occurring between time intervals of duration dt. Provided dt is sufficiently small such that at most one of the reactions $\gamma^{1:K}$ occurs, we need to consider only the changes induced by individual reactions. Formally, these are captured in

$$\tilde{P}_{\tilde{\boldsymbol{c}}}(t+dt) - \tilde{P}_{\tilde{\boldsymbol{c}}}(t) \approx \tilde{P}_{\tilde{\boldsymbol{c}}}(t)\left(\exp\left(-dt\sum_k \mu^k_{\tilde{\boldsymbol{c}}}\right) - 1\right) + \sum_k \tilde{P}_{\tilde{\boldsymbol{c}}-\tilde{\boldsymbol{\zeta}}^k}(t)\left(1 - \exp\left(-dt\mu^k_{\tilde{\boldsymbol{c}}-\tilde{\boldsymbol{\zeta}}^k}\right)\right)$$

Rearranging this equation, and taking the limit $dt \to 0^+$, we obtain the differential form of the master equation

$$\frac{d}{dt}\tilde{P}_{\tilde{\boldsymbol{c}}}(t) = -\left(\sum_k \mu^k_{\tilde{\boldsymbol{c}}}\right)\tilde{P}_{\tilde{\boldsymbol{c}}}(t) + \sum_k \mu^k_{\tilde{\boldsymbol{c}}-\tilde{\boldsymbol{\zeta}}^k}\tilde{P}_{\tilde{\boldsymbol{c}}-\tilde{\boldsymbol{\zeta}}^k}(t). \tag{2.17}$$

As this form is commonly found in chemical applications, it is most commonly called the *chemical master equation*.

2.3.4 A case study in chemical systems*

Modeling a chemical system

Composite systems are useful when modeling the joint states of different molecules in a chemical system and wishing to keep track of their individual states as a function of time. That is, retaining molecular identity.

*This is an advanced topic and could be skipped on a first reading.

Retaining molecular identity is important when we deal with molecules (such as DNA plasmids or mRNAs) present in small numbers whose behavior from molecule to molecule vary. For example, one DNA plasmid (of which there may be only a few replicate copies) may be transcriptionally inhibited for some time while another may be actively transcribing its DNA.

Retaining molecular identity however can quickly become computationally cumbersome if we are dealing with hundreds, thousands or more molecules. In this case, we may only be interested in modeling how a population of *chemical species* evolves without keeping track of the state of each individual molecule within each species. As such, the collapsed state formalism may be preferred simply on account of its reduced computational cost. Keeping track of and propagating in time the collapsed state (*i.e.* the population) of all species is strikingly less expensive than the state of each molecule. For example, in dealing when dealing with three chemical species, whose state-space is denoted with σ_1, σ_2 and σ_3, the collapsed state-space before the n^{th} jump is $\tilde{\boldsymbol{c}}_n = [c_n^{\sigma_1}, c_n^{\sigma_2}, c_n^{\sigma_3}]$ where each element coincides with the population of the superscripted species.

Note 2.15: Chemical species

The term "chemical species" is not uniquely defined and depends on the problem at hand. Typically, when we speak of different chemical species, we may be speaking molecules with a different number of atoms, types of atoms, or arrangements of bonds. However, it is possible, indeed common depending on the level of modeling, to refer to all RNA (irrespective of sequence and thus composition) as one species in order to distinguish it from protein (another species). At the other extreme, *i.e.* at a finer scale, it is even possible to speak of different species as subtly different electronic states of molecules.

The term chemical species also capture abstract differences that are difficult to describe in terms of atom numbers, bond arrangements or quantum states. For example, different chemical species may include actively transcribing DNA loci versus transcriptionally inhibited loci irrespective of how transcription or inhibition is achieved; *i.e.* what transcription factors may or may not be involved.

Finally, we draw a verbal, but not mathematical, distinction between configurational species and chemical species. For example, we mathematically treat different configurations of polymers (elongated versus compact) in the same way we treat chemical species.

Besides the reduction of computational cost, working with populations is especially critical when considering interacting molecules. For example, in constructing the composite transition rate matrix of two systems, such as two molecules, we previously wrote $\bar{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}^1 \oplus \boldsymbol{\Lambda}^2$. As all rates of double transitions in $\bar{\boldsymbol{\Lambda}}$ are zero, this pre-assumes that the molecules are non-interacting. Yet bimolecular reactions, where two molecules react to create a new molecule or a molecule dissociates releasing two molecules, cannot be feasibly treated within a composite formulation. Yet, as we will see, bimolecular and higher order molecular reactions are easily treated with a collapsed state-space formalism.

Example 2.14: Bimolecular reaction

As an illustrative example, we consider a simple *bimolecular reaction*

$$\mathrm{A} + \mathrm{B} \xrightarrow{\lambda_{+1}} \mathrm{AB}$$

where 1 molecule of species A and 1 molecule of species B collide, with a rate λ_{+1}, to form 1 molecule of species AB. For clarity, we assume that initially our system contains $J = 10$ molecules with 5 A's and 5 B's. That is, we take $\tilde{\boldsymbol{c}}_1 = [c_1^{\sigma_1}, c_1^{\sigma_2}, c_1^{\sigma_3}] = [5, 5, 0]$ where the elementary states are σ_1 for A, σ_2 for B and σ_3 for AB. Once a reaction occurs, we are left with $J = 9$ molecules, and $\tilde{\boldsymbol{c}}_2 = [4, 4, 1]$.

Similarly, if we consider the reverse reaction

$$\mathrm{AB} \xrightarrow{\lambda_{-1}} \mathrm{A} + \mathrm{B}$$

and initiate from $\tilde{\boldsymbol{c}}_1 = [0, 0, 5]$, say, after one reaction, we reach $\tilde{\boldsymbol{c}}_2 = [1, 1, 4]$.

Note 2.16: Propensities

Here we list the rules to compute the propensities of some common reactions

$$\begin{array}{lcl}
\gamma^1 & 0 \xrightarrow{\lambda_{\gamma^1}} \mathsf{A} & \mu^1_{\tilde{c}_n} = \lambda_{\gamma^1} \\
\gamma^2 & \mathsf{A} \xrightarrow{\lambda_{\gamma^2}} \emptyset & \mu^2_{\tilde{c}_n} = c_n^{\sigma_1} \lambda_{\gamma^2} \\
\gamma^3 & \mathsf{A} \xrightarrow{\lambda_{\gamma^3}} \mathsf{B} & \mu^3_{\tilde{c}_n} = c_n^{\sigma_1} \lambda_{\gamma^3} \\
\gamma^4 & \mathsf{A}+\mathsf{B} \xrightarrow{\lambda_{\gamma^4}} \mathsf{AB} & \mu^4_{\tilde{c}_n} = c_n^{\sigma_1} c_n^{\sigma_2} \lambda_{\gamma^4} \\
\gamma^5 & \mathsf{AB} \xrightarrow{\lambda_{\gamma^5}} \mathsf{A}+\mathsf{B} & \mu^5_{\tilde{c}_n} = c_n^{\sigma_3} \lambda_{\gamma^5}
\end{array}$$

where the populations of species A, B and AB are tabulated in $\tilde{c}_n = [c_n^{\sigma_1}, c_n^{\sigma_2}, c_n^{\sigma_3}]$, as in example 2.14.

A note on propensity is in order. We recognize that preceding the rate of the reaction is the number of molecules in all chemical species involved in the chemical reaction. It is for this reason that reaction γ^4, which involved 2 reactant molecules, contains the product $c_n^{\sigma_1} c_n^{\sigma_2}$ in the propensity.

Finally, we can also encode the net change in each population induced by the reactions in $\tilde{\zeta}^{1:5}$ through the following array

$$\begin{array}{c} \\ \gamma^1 \\ \gamma^2 \\ \gamma^3 \\ \gamma^4 \\ \gamma^5 \end{array} \begin{array}{c} \begin{array}{ccc} \sigma_1 & \sigma_2 & \sigma_3 \end{array} \\ \begin{pmatrix} +1 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & +1 & 0 \\ -1 & -1 & +1 \\ +1 & +1 & -1 \end{pmatrix} \end{array} = \begin{pmatrix} \tilde{\zeta}^1 \\ \tilde{\zeta}^2 \\ \tilde{\zeta}^3 \\ \tilde{\zeta}^4 \\ \tilde{\zeta}^5 \end{pmatrix}.$$

We make a final note on the propensity of reactions such as $\mathsf{A}+\mathsf{A} \xrightarrow{\lambda_{\gamma^6}} \mathsf{A}_2$. In this case we recall that the propensity depends on the number of molecules involved in the reaction. That is, all pairs and as such $\mu^6_{\tilde{c}_n} = (c_n^{\sigma_1}(c_n^{\sigma_1}-1)/2!)\,\lambda_{\gamma^6}$.

Following a similar logic, the propensity of a reaction such as $\mathsf{A}+\mathsf{A}+\mathsf{A} \xrightarrow{\lambda_{\gamma^7}} \mathsf{A}_3$ reads $\mu^7_{\tilde{c}_n} = (c_n^{\sigma_1}(c_n^{\sigma_1}-1)(c_n^{\sigma_1}-2)/3!)\,\lambda_{\gamma^7}$.

As interactions are common in chemical applications, we adjust our terminology from the collapsed state Markov jump processes to accommodate them. In particular, we consider a system of interacting molecules that are distributed among M species $\sigma_{1:M}$. Further, we consider K possible reactions $\gamma^{1:K}$ that may lead from one species to another and may involve a single *or* multiple molecules. Of course, unlike before, in $\gamma^{1:K}$ we now allow reactions that may alter the total population of molecule; that is, reactions that may not necessarily be elemental.

Bimolecular or higher order reactions are still described by Markov jump processes. The reason for this is that the probability of forming a species AB at the $(n+1)^{\text{th}}$ jump only depends on the number of its A and B components available after the n^{th} jump. However, the rationale for why higher order reactions (*e.g.* $\mathsf{A}+\mathsf{A}+\mathsf{A} \to \mathsf{A}_3$) are considered Markovian is somewhat more opaque. That is because reactions beyond second order are normally considered *effective* reactions. As such, these can, in principle, be broken down in terms of simpler bimolecular reactions except that the rate of such reactions may be so fast, as compared to any measurement timescale, that no bimolecular event would ever be observed. Thus $\mathsf{A}+\mathsf{A} \to \mathsf{A}_2$ followed by $\mathsf{A}_2+\mathsf{A} \to \mathsf{A}_3$ would be abbreviated as $\mathsf{A}+\mathsf{A}+\mathsf{A} \to \mathsf{A}_3$.

Finally, a physical note on propensities is in order. By invoking propensities dependent on the number of molecules available of each relevant species, we inherently make a *well-stirred* approximation. That is, we assume that molecules randomize in physical space on timescales far exceeding the largest supported propensity. The error of this approximation is difficult to bound if the number of molecular species is allowed to grow to high numbers or indefinitely (such as in a simple birth process) and the time between reactions can, in practice, shrink

to nil.

Simulating a chemical system

In a chemical setting, we avoid forming composite states $\bar{s}_n$. Instead, we describe our system directly with collapsed states $\tilde{\boldsymbol{c}}_n = [c_n^{\sigma_1}, \cdots, c_n^{\sigma_m}, \cdots, c_n^{\sigma_M}]$ that track down only the populations of the molecules of each species. Provided that the propensities $\mu^k_{\tilde{\boldsymbol{c}}_n}$ can be computed, we can readily simulate the transitions $\tilde{\boldsymbol{c}}_n \to \tilde{\boldsymbol{C}}_{n+1}$. The simulation scheme is summarized in algorithm 2.2 and follows the recipe provided in section 2.3.3 extended to cover bimolecular and higher order reactions.

Algorithm 2.1: The Gillespie simulation

Initialize time $t = t_0$ and the population of the species $\tilde{\boldsymbol{c}} = [c^{\sigma_1}, \cdots, c^{\sigma_M}]$. Use $\tilde{\boldsymbol{c}}$ to initialize the propensities of the reactions $(\mu^1, \cdots, \mu^K)$ and compute $\mu^* = \sum_k \mu^k$.
As long as $t < t_{\max}$, iterate the following

- Sample the period until the next event

$$H \sim \text{Exponential}\,(\mu^*)$$

- Sample the reaction triggering the next event

$$G|\tilde{\boldsymbol{c}} \sim \text{Categorical}_{\gamma^{1:K}}\left(\frac{\mu^1}{\mu^*}, \cdots, \frac{\mu^K}{\mu^*}\right)$$

- Update t according to h.
- Update $\tilde{\boldsymbol{c}}$, $(\mu^1, \cdots, \mu^K)$, and μ^* according to the reaction g that was sampled using the associated array of $\tilde{\boldsymbol{\zeta}}$.

This algorithm is termed the *Gillespie simulation* and it has been a cornerstone in the simulation of stochastic events, especially in complex chemical systems.

Mass action laws

Master equations describe probabilities of having $\tilde{\boldsymbol{c}}$ elements at some time t given some initial conditions. From these equations, we can derive *mass action laws* familiar across the Natural Sciences. Mass action laws describe how the average number $\langle c^{\sigma_m}(t)\rangle$ of elements of a species σ_m changes over time. Formally, this average is

$$\langle c^{\sigma_m}(t)\rangle = \sum_{\tilde{\boldsymbol{c}}} c^{\sigma_m} \tilde{P}_{\tilde{\boldsymbol{c}}}(t)$$

where the sum is taken over every achievable value of $\tilde{\boldsymbol{c}}$, *i.e.* all M-tuples of non-negative integers. Here $\tilde{P}_{\tilde{\boldsymbol{c}}}(t)$ satisfies the chemical master equation

$$\frac{d}{dt}\tilde{P}_{\tilde{\boldsymbol{c}}}(t) = -\left(\sum_k \mu^k_{\tilde{\boldsymbol{c}}}\right)\tilde{P}_{\tilde{\boldsymbol{c}}}(t) + \sum_k \mu^k_{\tilde{\boldsymbol{c}}-\tilde{\boldsymbol{\zeta}}^k}\tilde{P}_{\tilde{\boldsymbol{c}}-\tilde{\boldsymbol{\zeta}}^k}(t). \tag{2.18}$$

which we derived earlier, eq. (2.17).

We can use eq. (2.18) to obtain differential equations describing the evolution of $\langle c^{\sigma_m}(t)\rangle$ for each σ_m. For this, we first multiply eq. (2.18) by $c^{\sigma_m}(t)$ and sum over each achievable $\tilde{\boldsymbol{c}}$. We illustrate the above with an example on the birth-death process.

Example 2.15: Mass action laws for the birth-death process

To derive the mass action law for birth-death events, we start by writing the master equation for the single species A whose population c^{A} has a birth rate λ_b and death rate λ_d. It reads

$$\frac{d}{dt}\tilde{P}_{c^{\mathsf{A}}}(t) = -\lambda_b \tilde{P}_{c^{\mathsf{A}}}(t) - \lambda_d c^{\mathsf{A}} \tilde{P}_{c^{\mathsf{A}}}(t) + \lambda_b \tilde{P}_{c^{\mathsf{A}}-1}(t) + \lambda_d (c^{\mathsf{A}}+1)\tilde{P}_{c^{\mathsf{A}}+1}(t).$$

Multiplying both sides of the equation by c^{A} and summing over c^{A} yields

$$\begin{aligned}\frac{d}{dt}\sum_{c^{\mathsf{A}}=0}^{\infty} c^{\mathsf{A}}\tilde{P}_{c^{\mathsf{A}}}(t) = &-\lambda_b \sum_{c^{\mathsf{A}}=0}^{\infty} c^{\mathsf{A}}\tilde{P}_{c^{\mathsf{A}}}(t) - \lambda_d \sum_{c^{\mathsf{A}}=0}^{\infty} (c^{\mathsf{A}})^2 \tilde{P}_{c^{\mathsf{A}}}(t) \\ &+ \lambda_b \sum_{c^{\mathsf{A}}=0}^{\infty} c^{\mathsf{A}}\tilde{P}_{c^{\mathsf{A}}-1}(t) + \lambda_d \sum_{c^{\mathsf{A}}=0}^{\infty} c^{\mathsf{A}}(c^{\mathsf{A}}+1)\tilde{P}_{c^{\mathsf{A}}+1}(t),\end{aligned}$$

which we can immediately simplify to obtain

$$\begin{aligned}\frac{d}{dt}\langle c^{\mathsf{A}}(t)\rangle = &-\lambda_b \langle c^{\mathsf{A}}(t)\rangle - \lambda_d \langle \left(c^{\mathsf{A}}(t)\right)^2\rangle \\ &+ \lambda_b \sum_{c^{\mathsf{A}}=0}^{\infty} c^{\mathsf{A}}\tilde{P}_{c^{\mathsf{A}}-1}(t) + \lambda_d \sum_{c^{\mathsf{A}}=0}^{\infty} c^{\mathsf{A}}(c^{\mathsf{A}}+1)\tilde{P}_{c^{\mathsf{A}}+1}(t).\end{aligned} \tag{2.19}$$

The third and fourth terms on the right hand side of the above, which have not yet been simplified, are averages with respect to a probability taken at $c^{\mathsf{A}}-1$ or $c^{\mathsf{A}}+1$. In order to re-write these as averages over a probability in c^{A}, we define a new dummy index $\ell = c^{\mathsf{A}}-1$ or $\ell = c^{\mathsf{A}}+1$. For example, for the third term on the right hand side of eq. (2.19), we have

$$\begin{aligned}\lambda_b \sum_{c^{\mathsf{A}}=0}^{\infty} c^{\mathsf{A}}\tilde{P}_{c^{\mathsf{A}}-1}(t) &= \lambda_b \sum_{\ell=-1}^{\infty} (\ell+1)P_\ell(t) \\ &= \lambda_b + \lambda_b \sum_{\ell=-1}^{\infty} \ell P_\ell(t) \\ &= \lambda_b + \lambda_b \sum_{\ell=0}^{\infty} \ell P_\ell(t) \\ &= \lambda_b + \lambda_b \langle c^{\mathsf{A}}(t)\rangle\end{aligned}$$

where in the second line, we took $P_{-1}(t) = 0$ and in the last line we re-substituted the dummy index ℓ for c^{A}. Thus

$$\lambda_b \sum_{c^{\mathsf{A}}=0}^{\infty} c^{\mathsf{A}}\tilde{P}_{c^{\mathsf{A}}-1}(t) = \lambda_b + \lambda_b \langle c^{\mathsf{A}}(t)\rangle.$$

Substituting the expression above into eq. (2.19) and repeating the exercise for the fourth term, we obtain

$$\frac{d}{dt}\langle c^{\mathsf{A}}(t)\rangle = \lambda_b - \lambda_d \langle c^{\mathsf{A}}(t)\rangle$$

which is the mass action law for the birth-death process. From this mass action law, we can easily compute the steady state by setting the left hand side to zero and obtain

$$\langle c^{\mathsf{A}}(t)\rangle = \frac{\lambda_b}{\lambda_d}.$$

Figure 2.7 summarizes the difference between Gillespie simulations, master equations, and mass action laws in the study of birth-death processes. Starting from a zero initial population, we can see that:

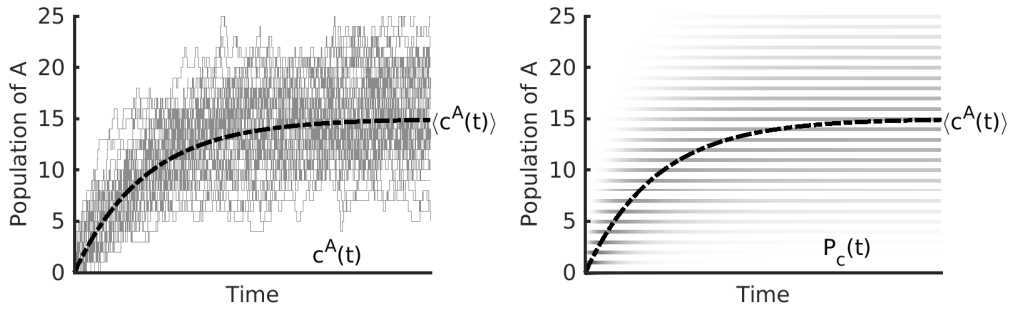


Figure 2.7: Here we show cartoon of a birth-death process where the birth rate exceeds the death rate such that the population increases until it reaches steady state. The left panel compares Gillespie trajectories with mass action; while, the right panel compares master equation with mass action.

- The Gillespie simulation captures individual birth and death events. The population $c^{\mathsf{A}}(t)$ is an integer valued variable that varies stochastically in time. Different stochastic realizations give rise to different trajectories. By contrast, the mass action law describes only the average population over every possible realization of $c^{\mathsf{A}}(t)$. The average population $\langle c^{\mathsf{A}}(t)\rangle$ is a continuous valued variable that changes deterministically across time.
- The master equation itself describes the probability of having a population $c^{\mathsf{A}}(t)$ at every time. Put differently, had we run an infinite number of Gillespie simulations and assessed the probability of each realization of $c^{\mathsf{A}}(t)$, from 0 to infinity, at each time, this probability would be the solution to the coinciding master equation.

For the birth-death process, as we saw, the mass action law becomes a differential equation of the form $\frac{d}{dt}\langle c^{\mathsf{A}}(t)\rangle = f(\langle c^{\mathsf{A}}(t)\rangle)$ where $f(\cdot)$ is a function of $\langle c^{\mathsf{A}}(t)\rangle$. In general, however, such closed form expressions may not exist. That is, depending on how the involved propensities, $\mu_{\tilde{\mathbf{c}}}^{1:K}$, depend on $\tilde{\mathbf{c}}$, we may very well obtain differential equations where the evolution of the average, $\frac{d}{dt}\langle c^{\sigma_m}(t)\rangle$, is related to the evolution of higher moments, such as $\langle (c^{\sigma_m}(t))^n\rangle$, or in cases with more than one species on mixed moments, such as $\left\langle (c^{\sigma_m}(t))^{n-\ell}\,(c^{\sigma_{m'}}(t))^{\ell}\right\rangle_{m\neq m'}$.

Inevitably, when the evolution of $\langle c^{\sigma_m}(t)\rangle$ depends on higher moments, we have non-closed form expressions and, to attain closed form expressions, we impose additional assumptions. For instance, suppose that $\frac{d}{dt}\langle c^{\sigma_m}(t)\rangle = f(\langle c^{\sigma_m}(t)\rangle, \langle (c^{\sigma_m}(t))^2\rangle)$ and that our goal is to obtain an expression where $f(\cdot,\cdot)$ has no dependency on $\langle (c^{\sigma_m}(t))^2\rangle$. In this case, a common assumption is to assume that second moments can be reduced to first moments squared, $\langle (c^{\sigma_m}(t))^2\rangle = \langle c^{\sigma_m}(t)\rangle^2$, and thus that variances, $\langle (c^{\sigma_m}(t))^2\rangle - \langle c^{\sigma_m}(t)\rangle^2$, are negligible as compared to means $\langle c^{\sigma_m}(t)\rangle$. On physical grounds, this is often more reasonable, say, than assuming that $\langle (c^{\sigma_m}(t))^2\rangle$ is small as compared to $\langle c^{\sigma_m}(t)\rangle$ as the number of chemical species present at any time, $c^{\sigma_m}(t)$, may perhaps always be large.

2.4 Systems with discrete state-spaces in discrete time

Here we consider modeling a system's state at *discrete* times only. This may happen either because the system itself inherently progresses in steps, for example as an idealization of reoccurring peaks of tidal waves, or as a simplification of a system inherently evolving in continuous time but assessed only at discrete steps. As we will see, formulations of discrete time systems are often highly artificial and, for their interpretation, we will frequently invoke *analogous* continuous time models.

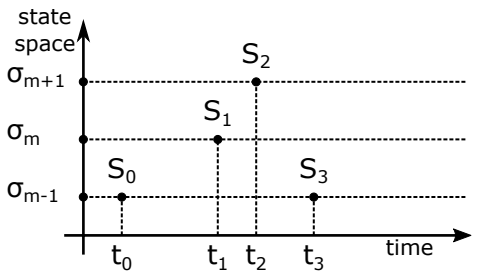


Figure 2.8: A system with a discrete state-space following discrete time dynamics. This system is described by passing states S_n at given times t_n.

2.4.1 Modeling a system at discrete times

To model a system in discrete time, fig. 2.8, we consider a *fixed* grid of time points $t_0, t_1, t_2, \ldots$. We require only that these time points be ordered such that they form a strictly increasing sequence

$$t_0 < t_1 < t_2 < \cdots < t_n < t_{n+1} < \cdots$$

Beyond such ordering, additional assumptions are unnecessary.

In most applications, of course, the grid of time points t_n is *regular*. That is, time points repeat at constant periods $t_{n+1} = t_n + \tau$, where τ is either set by a periodically occurring physical phenomenon or the acquisition period of individual measurements.

A system that evolves in discrete time is best described by the sequence of successive states

$$S_0 \to S_1 \to S_2 \to \cdots \to S_n \to S_{n+1} \to \cdots$$

that the system passes at *precisely* the grid time points $t_0, t_1, t_2, \ldots$. Since the system may be affected by random events, all $S_0, S_1, S_2, \ldots$ are random variables.

Unlike continuous time systems where we focused mostly on holding states, here we emphasize that S_n are *passing* states and that they may maintain the same value over successive steps. This may happen either because the underlying system does not change between successive time points t_n and t_{n+1} or because it changes and changes back again.

Note 2.17: What is a discrete-time trajectory?

The trajectory of a system with discrete state-space evolving in discrete time is a sequence $S_0, S_1, S_2, \ldots$.

Whenever our system results from the discretization of an analogous system that evolves in continuous time, in view of note 2.6, the passing states are related to the underlying continuous time trajectory $\mathcal{S}(\cdot)$ by $S_n = \mathcal{S}(t_n)$. As the trajectory $\mathcal{S}(\cdot)$ is random, the states S_n remain random too, despite being evaluated at fixed times.

2.4.2 Sampling a system at discrete times

As we focus on systems with discrete state-spaces, we will continue denoting constitutive states with $\sigma_{1:M}$. To complete our description, we need to specify an initialization rule for the selection of S_0 and the transition rules for the selection of $S_1, S_2, \ldots$.

Similar to systems evolving in continuous time, the initialization rule takes the form of a Categorical sampling

$$S_0 \sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\rho}) \tag{2.20}$$

where the probability array $\boldsymbol{\rho} = [\rho_{\sigma_1}, \ldots, \rho_{\sigma_M}]$ can model a preference for particular initializations. Under the Markov assumption, the transition rules also entail Categorical samplings

$$S_{n+1}|s_n \sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\pi}_{n,s_n}) \tag{2.21}$$

which, in the most general case, allow for transition probability arrays $\boldsymbol{\pi}_{n,\sigma_m} = [\pi_{n,\sigma_m\to\sigma_1}, \ldots, \pi_{n,\sigma_m\to\sigma_M}]$ that may be time dependent (and hence we have the added subscript n on the transition probability arrays).

Since transition rules must allow for self-transitions, $\pi_{n,\sigma_m\to\sigma_m}$ need not be zero. As such, transition probability matrices

$$\begin{array}{c} \\ \sigma_1 \\ \vdots \\ \sigma_M \end{array} \begin{array}{c} \begin{array}{ccc} \sigma_1 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \pi_{n,\sigma_1\to\sigma_1} & \cdots & \pi_{n,\sigma_1\to\sigma_M} \\ \vdots & \ddots & \vdots \\ \pi_{n,\sigma_M\to\sigma_1} & \cdots & \pi_{n,\sigma_M\to\sigma_M} \end{bmatrix} \end{array} = \boldsymbol{\Pi}_n$$

now assume *non-zero* diagonal elements. Finally, because our grid of time points in fixed, the notion of jump times or holding periods, that we encountered earlier, do not carry over. Accordingly, advancing our system is fully specified by eqs. (2.20) and (2.21) without additional random variables.

Note 2.18: The Markov chain

A sequence of states $S_0, S_1, S_2, \ldots$ sampled as described in eqs. (2.20) and (2.21) is termed *Markov chain*. We refer to a Markov chain as *inhomogeneous*, when the transition probability matrices $\boldsymbol{\Pi}_n$ differ from step to step. By contrast, we refer to a Markov chain as *homogeneous*, when the transition probability matrices remain the same for all steps. In the former case, we may denote them simply with $\boldsymbol{\Pi}$ and the corresponding arrays with $\boldsymbol{\pi}_{\sigma_m} = [\pi_{\sigma_m\to\sigma_1}, \ldots, \pi_{\sigma_m\to\sigma_M}]$.

2.4.3 Modeling kinetic schemes

To specify a kinetic scheme for our system, there are two major modeling approaches that we might adopt. We may choose to ascribe values directly to the transition probabilities $\pi_{n,\sigma_m\to\sigma_{m'}}$ or we may invoke an analogous system evolving in continuous time and derive these probabilities based on the kinetics of the underlying system. Below, we describe both approaches.

Ascribing transition probabilities

By specifying the values of each $\pi_{n,\sigma_m\to\sigma_{m'}}$, we have direct control on the allowed or disallowed transitions that the system may take. For example, by setting certain weights $\pi_{n,\sigma_m\to\sigma_{m'}}$ to zero, we can explicitly cancel particular transitions. Of course, each array $\boldsymbol{\pi}_{n,\sigma_m}$ must remain a probability array, so it must remain normalized to unity. This means that every $\boldsymbol{\pi}_{n,\sigma_m}$ contains at least one non-zero weight such that there always be at least one constitutive state available for sampling $S_{n+1}|s_n$.

Example 2.16: Random walks

In this example, we consider a system with $M = 5$ constitutive states that evolves according to a transition matrix

$$\boldsymbol{\Pi} = \begin{bmatrix} 1/5 & 4/5 & 0 & 0 & 0 \\ 2/5 & 1/5 & 2/5 & 0 & 0 \\ 0 & 2/5 & 1/5 & 2/5 & 0 \\ 0 & 0 & 2/5 & 1/5 & 2/5 \\ 0 & 0 & 0 & 4/5 & 1/5 \end{bmatrix}$$

that is fixed in time. In other words, our system is a homogenous Markov chain where each state may lead only to the state immediately above or immediately below. Such Markov chains are termed *random walks*.

As can be seen in fig. 2.9, once the system reaches the state-space's edges at σ_1 or σ_5, it bounces back.

If instead, we allow the system to exit and reenter from the other side of the state-space, the transition matrix

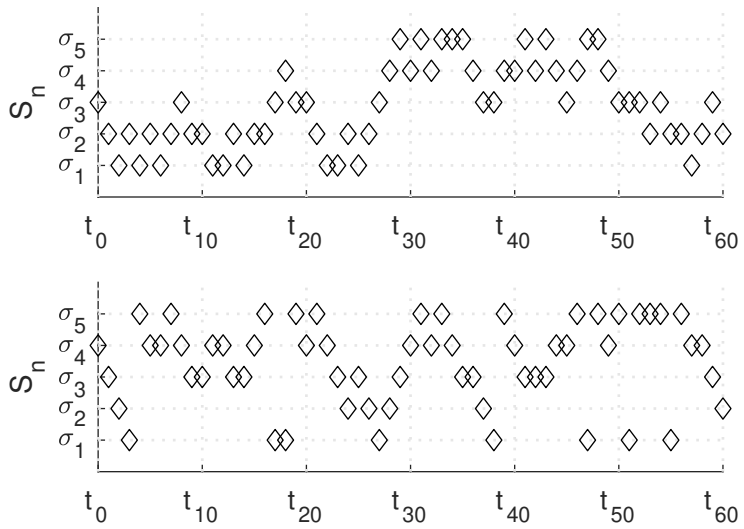


Figure 2.9: Two Markov chains representing random walks. In the upper panel the system bounces at the boundaries; while, in the lower panel the system exits and reenters from the other side.

becomes

$$\mathbf{\Pi} = \begin{bmatrix} 1/5 & 2/5 & 0 & 0 & 2/5 \\ 2/5 & 1/5 & 2/5 & 0 & 0 \\ 0 & 2/5 & 1/5 & 2/5 & 0 \\ 0 & 0 & 2/5 & 1/5 & 2/5 \\ 2/5 & 0 & 0 & 2/5 & 1/5 \end{bmatrix}.$$

Such Markov chains are termed *circular random walks*.

Ascribing transition rates

Directly ascribing the values of each $\pi_{n,\sigma_m \to \sigma_{m'}}$ might be cumbersome or, most importantly, might be highly demanding for physical systems, especially when the reference time points $t_0, t_1, t_2, \ldots$ are irregular. An alternative approach that avoids these pitfalls proceeds by starting our modeling with an analogous system that evolves in continuous time and uses its propagator to arrive at transition matrices for evolution in discrete time. We demonstrate this approach with an example.

Example 2.17: Discretizing a Markov jump process

In this example, we consider an analogous system in continuous time that is modeled by a Markov jump process. For this system, we consider a state-space $\sigma_{1:M}$ and a transition rate matrix $\mathbf{\Lambda}$. As we have seen in note 2.9, from $\mathbf{\Lambda}$ we can readily obtain the generator $\boldsymbol{G}$ and, through eq. (2.14), we can obtain its propagator $\boldsymbol{Q}^{t \to t'}$ relating the state probabilities at times t and t'.

According to eq. (2.21), each transition $s_n \to S_{n+1}$ entails a probability matrix $\mathbf{\Pi}_n$. Given that this matrix relates system states at times t_n and t_{n+1}, it is given by

$$\mathbf{\Pi}_n = \boldsymbol{Q}^{t_n \to t_{n+1}} = \mathbf{exp}\left((t_{n+1} - t_n)\boldsymbol{G}\right). \tag{2.22}$$

Since the matrix exponential, generally, does not have a closed form, in most cases the matrices $\mathbf{\Pi}_n$ need to be evaluated numerically, even when all rates contained within $\mathbf{\Lambda}$ are known.

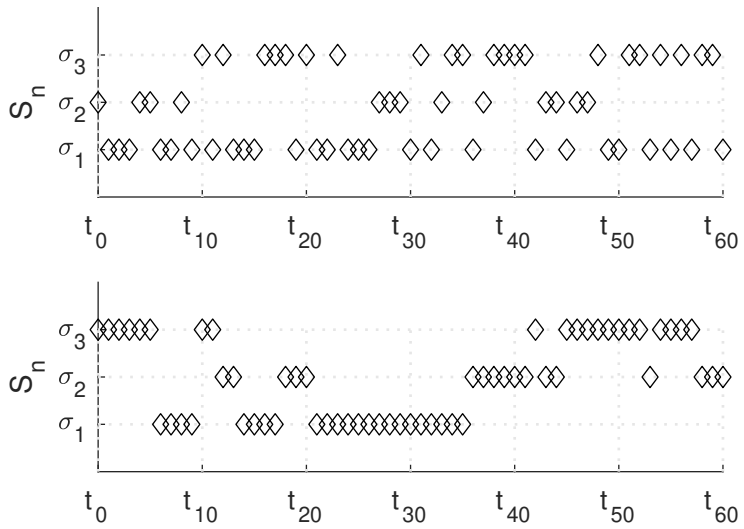


Figure 2.10: Two systems with different state persistences.

As a concrete example, we consider an underlying Markov jump processes with $M = 2$ constitutive states. Such a system is characterized by two rates $\lambda_{\sigma_1 \to \sigma_2}, \lambda_{\sigma_2 \to \sigma_1}$ and the rate matrix

$$\mathbf{\Lambda} = \begin{bmatrix} 0 & \lambda_{\sigma_1 \to \sigma_2} \\ \lambda_{\sigma_2 \to \sigma_1} & 0 \end{bmatrix}.$$

Assuming that such a Markov jump process is assessed at regular times $t_n = t_0 + n\tau$, the resulting transition probability matrices are related to the rates $\lambda_{\sigma_1 \to \sigma_2}$ and $\lambda_{\sigma_2 \to \sigma_1}$ by

$$\mathbf{\Pi} = \mathbf{exp}\left(\tau \begin{bmatrix} -\lambda_{\sigma_1 \to \sigma_2} & \lambda_{\sigma_1 \to \sigma_2} \\ \lambda_{\sigma_2 \to \sigma_1} & -\lambda_{\sigma_2 \to \sigma_1} \end{bmatrix}\right)$$

which is the same for all n steps. That is, owing to the lack of memory of the underlying Markov jump process and the regularity of the time grid, the resulting transition matrices are equal to each other.

Note 2.19: Interpreting transition probabilities

Starting with some transition probability matrix $\mathbf{\Pi}_n$, it is tempting to reverse eq. (2.22) to the generator $\boldsymbol{G}$ from which we may extract rates, $\lambda_{\sigma_m \to \sigma_{m'}}$.

For example, according to note 2.10, one approach is to approximate the matrix exponential $\mathbf{exp}(\ell \boldsymbol{L}) \approx \boldsymbol{I} + \ell \boldsymbol{L}$ and arrive at

$$\boldsymbol{G} \approx \frac{\mathbf{\Pi}_n - \boldsymbol{I}}{t_{n+1} - t_n}.$$

Although this approximation is formally valid provided $(t_{n+1} - t_n)\lambda_{\sigma_m \to \sigma_{m'}} \ll 1$ holds for all involved reactions, generally, such an approach is unsafe. Despite the errors it introduces in the analysis, it pre-assumes that the underlying dynamical system behaves as a Markov jump process. Indeed, a system in discrete time does *not* require such an underlying dynamical hypothesis.

2.4.4 Quantifying state persistence

For each $\mathbf{\Pi}_n$, self-transitions are weighted by $\pi_{n,\sigma_m\to\sigma_m}$. For "sticky" systems, that is systems with pronounced state persistence, self-transitions occur often. In these, $\pi_{n,\sigma_m\to\sigma_m}$ are the dominant weights in each $\boldsymbol{\pi}_{n,\sigma_m}$. Figure 2.10 illustrates two representative cases. The trajectories shown in the upper and lower panels are sampled with respective (fixed) matrices

$$\mathbf{\Pi} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}, \qquad \mathbf{\Pi} = \begin{bmatrix} 7/9 & 1/9 & 1/9 \\ 1/9 & 7/9 & 1/9 \\ 1/9 & 1/9 & 7/9 \end{bmatrix}.$$

As can be seen, the first system jumps to a different constitutive state at almost every step. By contrast, the second system typically stays in the same constitutive state for several steps before escaping.

Below, we show how to rigorously quantify state persistence in a given system. To begin, we suppose that the system reaches a constitutive state σ_m at some step n. In other words, $s_n = \sigma_m$. Further, we assume that $s_{n-1} \neq \sigma_m$, so the transition $s_{n-1} \to s_n$ initiates a holding period. This period may be terminated immediately, in which case it is followed by a transition $s_n \to S_{n+1} \neq \sigma_m$; or may be terminated after one step, in which case it is followed by $s_n \to S_{n+1} = \sigma_m \to S_{n+2} \neq \sigma_m$; or after two steps and so on.

To be more quantitative, we denote with D_{n,σ_m} the number of steps that our system needs in order to escape σ_m given that it enters σ_m at the n^{th} step. This is a random variable with values $0, 1, 2, \ldots$ that correspond to the following subsequences

$$\begin{aligned}
D_{n,\sigma_m} &= 0 \qquad \cdots \to s_{n-1} \neq \sigma_m \to s_n = \sigma_m \to s_{n+1} \neq \sigma_m \to \cdots \\
D_{n,\sigma_m} &= 1 \qquad \cdots \to s_{n-1} \neq \sigma_m \to s_n = \sigma_m \to s_{n+1} = \sigma_m \to s_{n+2} \neq \sigma_m \to \cdots \\
D_{n,\sigma_m} &= 2 \qquad \cdots \to s_{n-1} \neq \sigma_m \to s_n = \sigma_m \to s_{n+1} = \sigma_m \to s_{n+3} = \sigma_m \to s_{n+3} \neq \sigma_m \to \cdots
\end{aligned}$$

which we can use to derive precise probabilities. For instance, according to eq. (2.21), these probabilities are *proportional* to

$$\begin{aligned}
D_{n,\sigma_m} &= 0 \qquad & (1 - \pi_{n,\sigma_m\to\sigma_m}) \\
D_{n,\sigma_m} &= 1 \qquad & \pi_{n,\sigma_m\to\sigma_m}(1 - \pi_{n+1,\sigma_m\to\sigma_m}) \\
D_{n,\sigma_m} &= 2 \qquad & \pi_{n,\sigma_m\to\sigma_m}\pi_{n+1,\sigma_m\to\sigma_m}(1 - \pi_{n+2,\sigma_m\to\sigma_m})
\end{aligned}$$

from which we readily deduce the general expression

$$P_{n,\sigma_m}(d) = \left(\prod_{n'=n}^{n+d-1} \pi_{n',\sigma_m\to\sigma_m}\right)(1 - \pi_{n+d,\sigma_m\to\sigma_m}).$$

For homogenous Markov chains, the transition probabilities are fixed and $P_{n,\sigma_m}(d)$ as well as D_{n,σ_m} become independent of n. In particular, $P_{n,\sigma_m}(d)$ takes the form

$$P_{\sigma_m}(d) = (\pi_{\sigma_m\to\sigma_m})^d\,(1 - \pi_{\sigma_m\to\sigma_m})$$

which indicates that $D_{\sigma_m} \sim$ Geometric$(1-\pi_{\sigma_m\to\sigma_m})$. We can conclude that once our system enters a constitutive state σ_m, on average, it stays for additional $\pi_{\sigma_m\to\sigma_m}/(1 - \pi_{\sigma_m\to\sigma_m})$ steps.

Note 2.20: Interpreting dwell times

The notion of a *dwell* or *holding time* for a system in discrete time lacks physical meaning. As seen from $P_{n,\sigma_m}(d)$ and $P_{\sigma_m}(d)$, we can *only* quantify the statistics of steps d needed to escape a constitutive state. Of course, we may invoke the reference grid of time points t_n and always interpret these statistics in temporal terms. For example, with a homogenous Markov chain at regular time points $t_n = t_0 + n\tau$, we may transform the mean number of steps needed to escape a state to a mean dwell time $\tau\pi_{\sigma_m\to\sigma_m}/(1 - \pi_{\sigma_m\to\sigma_m})$. However, we need to be careful

with our definitions, as what is meant by "time" or "period", in the context of a discrete system, is ambiguous.

2.5 Systems with continuous state-spaces in discrete time

We now turn to systems whose state-spaces form a continuum. In this section, we first consider modeling such systems in discrete time steps; that is, only at a fixed grid of time points.

Throughout, we will exclusively consider a fixed grid of reference time points $t_0, t_1, t_2, \cdots$. As before, we will treat this grid as time ordered with $t_n < t_{n+1}$, although not necessarily regular. For clarity, we will denote with $R_0, R_1, R_2, \cdots$ the states through which the system passes at precisely those times.

Each R_n will be treated as a random variable and attains continuous values. For this reason, Categorical distributions that we have used so far are inadequate in formulating initialization and transition rules. Instead, these rules must now be provided by probability distributions that allow for continuous variables. This requires that our sampling schemes be determined by probability densities adapted to our specific state-spaces.

To describe the initialization rule, for R_0 we need to quantify the probability

$$\text{Probability of } r_0 \text{ being } dr \text{ near } r = \rho(r)dr$$

which we may summarize into

$$R_0 \sim \mathbb{R} \tag{2.23}$$

for some appropriate distribution $\mathbb{R}$ with density $\rho(r)$. Similarly, to describe the transition rules, for each $R_n \to R_{n+1}$ we need to quantify the probability

$$\text{Probability of } r_{n+1} \text{ being } dr \text{ near } r \text{ given } r_n = \pi_n(r|r_n)dr$$

described by

$$R_{n+1}|r_n \sim \mathbb{P}_n(r_n) \tag{2.24}$$

for appropriate distributions with density $\pi_n(r|r')$. These are the direct analogs of eqs. (2.20) and (2.21) for the discrete state-space systems discussed earlier. The densities $\rho(r)$ and $\pi(r|r')$ are normalized with respect to r, *i.e.* $\int_r dr\, \rho(r) = 1$ and $\int_r dr\, \pi_n(r|r') = 1$.

The exact form of $\rho(r)$ and $\pi_n(r|r')$ depends on the model describing the physical system at hand and over the next sections we will explore some common cases. As we will see, the description of the system is often significantly simplified for transition rules dictated by random walks. In such cases, for each transition $r_n \to R_{n+1}$, instead of modeling directly the state variable R_{n+1}, we instead model an increment dR_n and *derive* $r_{n+1} = r_n + dr_n$. Of course, both descriptions are equivalent and we often use them interchangeably.

2.5.1 Mechanical systems with state independent forces

We begin by discussing systems with no stochasticity which are much simpler to formulate. We begin by considering the state of the system at some time t given by $\boldsymbol{r}(t) = (q(t), p(t))$, where $q(t)$ is a position and $p(t)$ a momentum variable in a continuous state-space that is termed *phase-space*. For example, a point mass subject to a force $F(t)$ that does *not* vary spatially, according to Newton's laws evolves according to

$$dq(t) = \frac{p(t)}{m}dt, \tag{2.25}$$

$$dp(t) = F(t)dt \tag{2.26}$$

where, we have written the dt on the right hand side of eqs. (2.25) and (2.26). This notation will be convenient later as we introduce Brownian motion (later in eq. (2.32)) and describe random forces acting during a time interval around t of duration dt.

Note 2.21: Differential equations

Equations (2.25) and (2.26) are equivalent to the more familiar

$$\dot{q}(t) = \frac{p(t)}{m}, \tag{2.27}$$
$$\dot{p}(t) = F(t) \tag{2.28}$$

where $\dot{q}$ and $\dot{p}$ stand for the temporal derivatives dq/dt and dp/dt, respectively.

Of course, similar to all descriptions mediated by differential equations, neither eqs. (2.25) and (2.26), nor eqs. (2.27) and (2.28), have a stand-alone meaning. Instead, both sets are interpreted as informal ways of expressing the corresponding integral forms

$$\int_{t'}^{t''} dq = \int_{t'}^{t''} dt\, \frac{p(t)}{m},$$
$$\int_{t'}^{t''} dp = \int_{t'}^{t''} dt\, F(t)$$

that are more cumbersome to spell out. These integral forms, which should hold for any meaningful choice of t' and t'', can also be written as

$$q(t'') - q(t') = \int_{t'}^{t''} dt\, \frac{p(t)}{m},$$
$$p(t'') - p(t') = \int_{t'}^{t''} dt\, F(t).$$

At this point, we want to emphasize that mathematical models expressed in terms of differential equations, *always involve a notion of integration*, even when integrals are only implicit.

By defining $\boldsymbol{r}_n = \boldsymbol{r}(t_n)$ at our reference time points t_n, we now show how we can use equations of motion to relate $\boldsymbol{r}_0, \boldsymbol{r}_1, \boldsymbol{r}_2, \cdots$ across time.

In particular, starting at a reference time point t_n and integrating up to some later time t, we obtain

$$q(t) = q_n + \frac{t - t_n}{m} p_n + \frac{1}{m} \int_{t_n}^{t} dt' \int_{t_n}^{t'} dt''\, F(t''),$$
$$p(t) = p_n + \int_{t_n}^{t} dt'\, F(t').$$

Applied at $t = t_{n+1}$, these lead to the transition rules

$$q_{n+1} = q_n + \frac{t_{n+1} - t_n}{m} p_n + q_n^*,$$
$$p_{n+1} = p_n + p_n^*.$$

where $q_n^* = \frac{1}{m} \int_{t_n}^{t_{n+1}} dt' \int_{t_n}^{t'} dt''\, F(t'')$ and $p_n^* = \int_{t_n}^{t_{n+1}} dt'\, F(t')$ are constants unrelated to the state variables $\boldsymbol{r}_0, \boldsymbol{r}_1, \boldsymbol{r}_2, \cdots$. Equivalently, we may expresses the transition rules in terms of increments

$$dq_n = \frac{t_{n+1} - t_n}{m} p_n + q_n^*,$$
$$dp_n = p_n^*.$$

While the evolution of these systems is inherently deterministic, uncertainty may stem from the initial conditions $(q_0, p_0) \sim \mathbb{R}$. Once initial conditions are sampled, the (deterministic) transitions rules can be seen as a singular

case of eq. (2.24) as

$$Q_{n+1}|(q_n,p_n) \sim \delta_{q_n+\frac{t_{n+1}-t_n}{m}p_n+q_n^*},$$
$$P_{n+1}|(q_n,p_n) \sim \delta_{p_n+p_n^*}.$$

2.5.2 Mechanical systems with state dependent forces

According to Newton's laws, a point mass subject to a general force evolves according to

$$dq(t) = \frac{p(t)}{m}dt,$$
$$dp(t) = F(t,\boldsymbol{r}(t))dt.$$

The critical difference is that now the force $F(t,\boldsymbol{r}(t))$ now also depends on the state $\boldsymbol{r}(t)$.

Just as with state independent forces, we will derive appropriate transition rules relating the states $\boldsymbol{r}_0,\boldsymbol{r}_1,\boldsymbol{r}_2,\cdots$ across time. In particular, starting at some t_n and integrating up to some later time t, we obtain

$$q(t) = q_n + (t-t_n)\frac{p_n}{m} + \frac{1}{m}\int_{t_n}^{t} dt' \int_{t_n}^{t'} dt''\, F\left(t'',\boldsymbol{r}(t'')\right),$$
$$p(t) = p_n + \int_{t_n}^{t} dt'\, F\left(t',\boldsymbol{r}(t')\right).$$

Provided $F\left(t,\boldsymbol{r}(t)\right)$ does not change considerably between t_n and t, we may assume

$$F\left(t,\boldsymbol{r}(t)\right) \approx F\left(t_n,\boldsymbol{r}_n\right).$$

Under this approximation, we readily obtain

$$q(t) \approx q_n + (t-t_n)\frac{p_n}{m} + \frac{(t-t_n)^2}{2m}F\left(t_n,\boldsymbol{r}_n\right),$$
$$p(t) \approx p_n + (t-t_n)F\left(t_n,\boldsymbol{r}_n\right),$$

which, applied at $t = t_{n+1}$ and ignoring the approximation, leads to the transition rules

$$q_{n+1} = q_n + (t_{n+1}-t_n)\frac{p_n}{m} + \frac{(t_{n+1}-t_n)^2}{2m}F\left(t_n,\boldsymbol{r}_n\right),$$
$$p_{n+1} = p_n + (t_{n+1}-t_n)F\left(t_n,\boldsymbol{r}_n\right).$$

Equivalently, we may expresses the transition rules in terms of increments

$$dq_n = (t_{n+1}-t_n)\frac{p_n}{m} + \frac{(t_{n+1}-t_n)^2}{2m}F\left(t_n,\boldsymbol{r}_n\right), \tag{2.29}$$
$$dp_n = (t_{n+1}-t_n)F\left(t_n,\boldsymbol{r}_n\right). \tag{2.30}$$

If we can further assume that t_{n+1} and t_n are fairly close to each other, then we may further approximate $(t_{n+1}-t_n)^2 \approx 0$. In which case, we end up with the transition rules

$$q_{n+1} = q_n + (t_{n+1}-t_n)\frac{p_n}{m} \qquad dq_n = (t_{n+1}-t_n)\frac{p_n}{m}$$
$$p_{n+1} = p_n + (t_{n+1}-t_n)F\left(t_n,\boldsymbol{r}_n\right), \qquad dp_n = (t_{n+1}-t_n)F\left(t_n,\boldsymbol{r}_n\right).$$

Note 2.22: Temporal discretization

To derive the transitions rules we had to invoke certain *approximations*. Initially, we assumed $F(t, \boldsymbol{r}(t)) \approx F(t_n, \boldsymbol{r}_n)$ holds for any t between t_n and t_{n+1}; while, we additionally assumed $(t_{n+1} - t_n)^2 \approx 0$ and derived a different rule.

Specific approximations, or numerical schemes, applied to the underlying equation of motion determine:

- when the rule is valid or invalid,
- the resulting rule's functional form.

While different approximations, for instance $F(t, \boldsymbol{r}(t)) \approx F(t_{n+1}, \boldsymbol{r}_{n+1})$ or even $F(t, \boldsymbol{r}(t)) \approx \frac{1}{2}(F(t_n, \boldsymbol{r}_n) + F(t_{n+1}, \boldsymbol{r}_{n+1}))$, may lead to more accurate transition rules, they introduce a more complex relationship between $\boldsymbol{r}_n$ to $\boldsymbol{r}_{n+1}$. The accuracy gained must thus be weighed against its adverse effect on the simplicity of an eventual inverse scheme.

Finally, we emphasize an important and subtle point. Time discretized equations are never the same as their continuous time formulation. Here Exercise 2.4 uses a simple harmonic oscillator to illustrate such side effects on an important quantity: the energy of the system along its trajectories.

2.5.3 Langevin dynamics

We are motivated by thermal or other sources of uncertainty to prescribe specific forms for the transition densities. In this case, the equation of motion satisfied by a point mass is subject to a general force, as before, supplemented by a fluctuating component, often colloquially termed *white noise*.

This fluctuating force models the effect imparted by collisions (more precisely electrostatic and other interactions) of unobserved particles contained in the medium in which the point mass is embedded. In practice, particles of the medium are often understood as being solvent molecules or other point masses interacting with the point mass of interest not explicitly modeled.

Thermal physics dictates that if the fluctuating force provides an energy source, then there must also exist an energy sink. That is, the point particle must also be subject to friction.

Dynamics with fluctuating forces and the Langevin equation

One of the simplest dynamical models according to which the point mass may evolve is a *Langevin model* or, put differently, we say the point mass satisfies Langevin dynamics. This model consists of the following equations of motion

$$dq(t) = \frac{p(t)}{m} dt, \tag{2.31}$$

$$dp(t) = F(t, \boldsymbol{r}(t))dt - \zeta \frac{p(t)}{m} dt + B\, dW_t \tag{2.32}$$

where ζ is the particle's friction coefficient, and B dictates the strength of the force fluctuations, dW_t. The latter are defined only indirectly by the characteristic property

$$\int_{t'}^{t''} dW_t \sim \text{Normal}\,(0, |t'' - t'|) \tag{2.33}$$

which holds for any instances of time t' and t''. Formally, this means that the fluctuations dW_t integrated over a time interval, between some t' and t'', form a random variable with normal statistics of zero mean and variance equal to $|t'' - t'|$. Most of the times, it is more convenient to express such accumulated random increments by $\int_{t'}^{t''} dW_t = |t'' - t'|^{1/2} \xi_{t',t''}$ where $\xi_{t',t''} \sim \text{Normal}(0, 1)$.

To derive transition densities under the Langevin model, just as before, we integrate position once, and ignoring the approximation, obtain

$$q_{n+1} = q_n + (t_{n+1} - t_n)\frac{p_n}{m}.$$

However, now we need to be a little more careful with the integration of momentum which reads

$$p_{n+1} = p_n + \int_{t_n}^{t_{n+1}} dt'\, F\left(t', \boldsymbol{r}(t')\right) - \frac{\zeta}{m}\int_{t_n}^{t_{n+1}} dt'\, p(t') + B\int_{t_n}^{t_{n+1}} dW_t.$$

The second and third term on the right hand side of the above are approximated as before assuming $F\left(t, \boldsymbol{r}(t)\right)$ and $p(t)$ vary negligibly under integration. However, the third term of the right hand side is integrated exactly and, according to eq. (2.33), yields $B(t_{n+1} - t_n)^{1/2}\xi_n$ where $\xi_n \sim \text{Normal}(0, 1)$. Putting it all together, we have

$$\begin{aligned} p_{n+1} &= p_n\left(1 - \frac{\zeta}{m}(t_{n+1} - t_n)\right) + (t_{n+1} - t_n)F\left(t_n, \boldsymbol{r}(t_n)\right) + B(t_{n+1} - t_n)^{1/2}\xi_n, \\ \xi_n &\sim \text{Normal}(0, 1). \end{aligned}$$

In terms of eq. (2.24), the Langevin transition densities can be summarized as

$$Q_{n+1}|(q_n, p_n) \sim \delta_{q_n + (t_{n+1} - t_n)\frac{p_n}{m}} \tag{2.34}$$

$$P_{n+1}|(q_n, p_n) \sim \text{Normal}\left(p_n\left(1 - \frac{\zeta}{m}(t_{n+1} - t_n)\right) + (t_{n+1} - t_n)F\left(t_n, \boldsymbol{r}_n\right), B^2(t_{n+1} - t_n)\right). \tag{2.35}$$

The physics behind the Langevin equation*

In the Natural Sciences, the Langevin equation, explicitly understood in three dimensions here, is often written as

$$\dot{q} = \frac{p}{m} \tag{2.36}$$

$$\dot{p} = -\nabla U(q) - \zeta\frac{p}{m} + F_p(t) \tag{2.37}$$

where we use $\dot{q}$ and $\dot{p}$ to denote the time derivatives dq/dt and dp/dt, respectively. We also replace the non-fluctuating portion of the force by the gradient of a potential as it is often customary to have conservative forces in physical applications.

Here $F_p(t)$ is an instantaneous force that changes randomly over time, with one such force for each momentum in all 3 dimensions. This force changes so violently (discontinuously) that it is worth considering $F_p(t)$ when integrated over a finite period of time. In particular, between some instances of time t' and t'', the accumulated force $\int_{t'}^{t''} dt\, F_p(t)$ is a random variable which, by definition, obeys Normal statistics of zero mean and variance *inversely* proportional to $|t'' - t'|$. As such, we write

$$\int_{t'}^{t''} dt\, F_p(t) \sim \text{Normal}\left(0, \frac{2B^2}{|t'' - t'|}\right). \tag{2.38}$$

or $\int_{t'}^{t''} dt\, F_p(t) = \sqrt{2}B|t'' - t'|^{-1/2}\xi_{t',t''}$ where $\xi_{t',t''} \sim \text{Normal}(0, 1)$. Often, instead of eq. (2.38), the description of the fluctuating force is abbreviated as

$$\langle F_p(t)\rangle = 0, \qquad \langle F_p(t) \cdot F_p(t')\rangle = 3 \cdot 2B^2\delta_{t'}(t). \tag{2.39}$$

The factor of 3 on the right hand side of the above coincides with the number of dimensions and arises from the contraction of a 3-component vector of $F_p(t)$. The δ-function appearing on the right hand side of eq. (2.39) suggests that the white noise here is *time-decorrelated*.

While intuitive, eqs. (2.36) and (2.37) are mathematically problematic. On the one hand, the left hand side of the equation, $\dot{p}$, is to be interpreted in continuous time while it is unclear how to interpret $F_p(t)$ in continuous time without a surrounding integral.

*This is an advanced topic and could be skipped on a first reading.

The Langevin equation of eqs. (2.36) and (2.37) can be made to look like the more familiar Newton's second law by taking a time derivative of $\dot{q}$ in eq. (2.37) and inserting into this expression, the formula for $\dot{p}$

$$m\ddot{q} = -\nabla U(q) - \zeta\dot{q} + F_p(t). \tag{2.40}$$

Equation (2.40) is sometimes termed the *underdamped Langevin equation* as it contains the inertial mass of the point particle.

We now highlight some assumptions inherent in eq. (2.40) or, equivalently, eqs. (2.36) and (2.37). First, as written in eq. (2.40), this Langevin equation assumes a classical dynamical paradigm and models the diffusion of the particle in a scalar potential $U(q)$ and otherwise homogeneous and isotropic medium. The diffusion coefficient, to be related to the friction coefficient shortly, is assumed independent of space. Perhaps less intuitive is the assumption that the solvent in which the point particle is embedded, which drives both dissipative and fluctuating forces, instantaneously re-adjusts to the position of the particle, of effective inertial mass m, causing a constant drag on the particle, ζ and $F_p(t)$.

Physics further dictates a relationship between ζ and B as a particle cannot indefinitely accumulate energy (if dissipation is too weak) nor can its average kinetic energy drop to zero (if dissipation is too strong) as equilibrium statistical physics insists that the average translation kinetic energy in 3 dimensions be $3k_BT/2$ where T is the equilibrium temperature and k_B is Boltzmann's constant. That is, ζ and B are related to each other through an equation called the fluctuation dissipation theorem and both are related to the temperature of the solvent.

Example 2.18: Fluctuation dissipation theorem

To show the fluctuation dissipation theorem, for the simple case of no potential we use the integrating factor method to re-write eq. (2.37) as follows

$$p(t) = e^{-\frac{\zeta}{m}t}p(0) + \int_0^t ds\, e^{-\frac{\zeta}{m}(t-s)}F_p(s) \tag{2.41}$$

or, simply,

$$\frac{p(t)}{m} = v(t) = \frac{dq}{dt} = e^{-\frac{\zeta}{m}t}\frac{dq(0)}{dt} + F_q(t) \tag{2.42}$$

where $v(t)$ is the velocity and

$$F_q(t) = \frac{1}{m}\int_0^t ds\, e^{-\frac{\zeta}{m}(t-s)}F_p(s). \tag{2.43}$$

It is understood that $F_q(t)$ has units of velocity.

At equilibrium, we know from statistical physics that $\frac{m}{2}v_{eq}^2 \equiv \lim_{t\to\infty}\frac{m}{2}\langle v^2(t)\rangle = 3k_BT/2$ where the average is over all realizations of the noise.

Taking the square of eq. (2.42) and averaging over noise by exploiting eq. (2.39) yields

$$\langle v^2(t)\rangle = \langle v(t)\cdot v(t)\rangle = \left(e^{-\frac{\zeta}{m}t}\frac{dq(0)}{dt}\right)^2 + \langle F_q(t)\cdot F_q(t)\rangle.$$

The final term can be written as

$$\begin{aligned}\langle F_q(t)\cdot F_q(t)\rangle &= \frac{6B^2}{m^2}\int_0^t ds'\int_0^t ds\, e^{-\frac{\zeta}{m}(2t-s-s')}\delta_{s'}(s)\\ &= \frac{6B^2}{m^2}\int_0^t ds\, e^{-\frac{2\zeta}{m}(t-s)}\\ &= \frac{6B^2}{2m\zeta}(1-e^{-2t\zeta/m}).\end{aligned} \tag{2.44}$$

Taking the $t \to \infty$ limit yields

$$\lim_{t\to\infty} \frac{m}{2}\langle v^2(t)\rangle = \frac{m}{2}\frac{6B^2}{2m\zeta} \qquad \text{thus} \qquad B^2 = \zeta k_B T \tag{2.45}$$

where we readily verify that both sides of the equality $B^2 = \zeta k_B T$, which is an expression of the fluctuation-dissipation theorem for our simplified case, have units of force squared. Put differently

$$\langle F_p(t) \cdot F_p(t')\rangle = 6B^2\delta_{t'}(t) = 6\zeta k_B T \delta_{t'}(t). \tag{2.46}$$

Here is another bit of Physics that simplifies the Langevin equation. Physics tells us that the fluctuating force acting on the momentum, expressed through the force $F_p(t)$, quickly randomizes the particle's momentum in time. It is for this reason that we loosely argue that "inertial effects do not matter" and the Langevin equation we most often see, termed the *overdamped Langevin equation*, is written as

$$\zeta\frac{dq}{dt} \approx -\nabla U(q) + F_p(t). \tag{2.47}$$

The justification is given below.

Example 2.19: Dropping inertia

From eq. (2.41) we see that the momentum at time t, $p(t)$, has exponentially decaying dependence on its initial condition, $p(0)$. In other words, momenta are randomized on m/ζ timescales.

Here we re-write eqs. (2.42) and (2.43) more generally in the presence of a potential

$$\frac{dq(t)}{dt} = e^{-\frac{\zeta}{m}t}\frac{dq(0)}{dt} - \frac{1}{m}\int_0^t ds\, e^{-\frac{\zeta}{m}(t-s)}\nabla U(q(s)) + F_q(t)$$
$$F_q(t) = \frac{1}{m}\int_0^t ds\, e^{-\frac{\zeta}{m}(t-s)}F_p(s).$$

Since ζ/m is interpreted as a large rate constant (*i.e.* inverse time) related to a rate of momentum relaxation, under these assumptions, the above simplify to

$$\zeta\frac{dq}{dt} \approx -\nabla U(q) + F_p(t). \tag{2.48}$$

A comparison of eq. (2.48) and eq. (2.37) reveals that, in this limit,

$$\dot{p} \approx 0.$$

Now that we have related both B and ζ to temperature through the fluctuation theorem, it is also worth relating these quantities to another quantity that appears in forward models across disciplines: the diffusion coefficient, D. The diffusion coefficient is often computed from the mean square displacement

$$6D = \frac{\langle (q(t_{n+1}) - q(t_n))^2\rangle}{t_{n+1} - t_n} \tag{2.49}$$

and this relation is sufficient, as we show below, to relate D to B and ζ.

Example 2.20: Mean square displacement

To relate the diffusion coefficient, D, to B we consider an isotropic homogeneous medium where D is defined through eq. (2.49).

We start by computing the right hand side of eq. (2.49) by considering eq. (2.48), assuming no potential

$(U = 0)$ and integrating once

$$q(t_{n+1}) - q(t_n) = \frac{1}{\zeta}\int_{t_n}^{t_{n+1}} ds\, F_p(s). \tag{2.50}$$

Taking the square of both sides and averaging over noise yields

$$\begin{aligned}\langle (q(t_{n+1}) - q(t_n))^2\rangle &= \frac{1}{\zeta^2}\int_{t_n}^{t_{n+1}} ds \int_{t_n}^{t_{n+1}} ds' \langle F_p(s)\cdot F_p(s')\rangle \\ &= \frac{6B^2}{\zeta^2}\int_{t_n}^{t_{n+1}} ds \int_{t_n}^{t_{n+1}} ds' \delta_{s'}(s) = \frac{6B^2(t_{n+1}-t_n)}{\zeta^2}.\end{aligned}$$

Since $\langle (q(t_{n+1}) - q(t_n))^2\rangle/(t_{n+1}-t_n) = 6D$, we have

$$6D = \frac{6B^2}{\zeta^2} \tag{2.51}$$

and, since from eq. (2.46) we have $B^2 = \zeta k_B T$, then

$$D = \frac{k_B T}{\zeta}.$$

We end with note on free Brownian motion in Physics. The overdamped motion of a freely (no potential) particle diffusing is termed *Brownian motion*. In the language of Physics, considering the overdamped case as we did in eq. (2.48), and setting $U = 0$, we have

$$\dot{q} = \frac{1}{\zeta}F_p(t) \tag{2.52}$$

where $F_p(t)$ is given by eq. (2.38). Integrating eq. (2.52) once on both sides over the time interval from t_n to t_{n+1}, we obtain, for each dimension,

$$q_{n+1} - q_n = \frac{1}{\zeta}\int_{t_n}^{t_{n+1}} dt F_p(t) \tag{2.53}$$

which, using eq. (2.38), simplifies to

$$q_{n+1} = q_n + \frac{\sqrt{2}B}{\zeta}(t_{n+1}-t_n)^{1/2}\xi_n. \tag{2.54}$$

Using eq. (2.51), in terms of the diffusion coefficient D, the equation satisfied by a particle's position is then $q_{n+1} = q_n + \sqrt{2D}(t_{n+1}-t_n)^{1/2}\xi_n$.

2.5.4 Models involving Brownian motion in discrete time

In the last section, we related the diffusion coefficient, D, to quantities derived earlier and arrived at $2D = B^2/\zeta^2$ from fundamental statistical physics considerations.

We then arrived at an equation of motion satisfied by a freely diffusing particle's position which reads $q_{n+1} = q_n + \sqrt{2D}(t_{n+1}-t_n)^{1/2}\xi_n$.

Another way to write this, is to say that the particle's position, $Q_n = (X_n, Y_n, Z_n)$, with respect to some Cartesian frame is sampled according to

$$\begin{aligned}X_{n+1}|x_n &\sim \text{Normal}(x_n, 2D(t_{n+1}-t_n)) \\ Y_{n+1}|y_n &\sim \text{Normal}(y_n, 2D(t_{n+1}-t_n)) \\ Z_{n+1}|z_n &\sim \text{Normal}(z_n, 2D(t_{n+1}-t_n)).\end{aligned}$$

In most applications, the initialization rule is unimportant, so we often omit it.

When we speak of a general state, $\boldsymbol{R}_n$, we say that it satisfies free Brownian motion (though the word free is often omitted) when

$$\boldsymbol{R}_{n+1}|\boldsymbol{r}_n \sim \text{Normal}_3\left(\boldsymbol{r}_n, 2D(t_{n+1}-t_n)\boldsymbol{I}\right). \tag{2.55}$$

Equivalently, Brownian motion can also be described as a random walk

$$d\boldsymbol{R}_n \sim \text{Normal}_3(\boldsymbol{0}, 2D(t_{n+1}-t_n)\boldsymbol{I}).$$

We recover the states by $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + d\boldsymbol{r}_n$.

Algorithm 2.2: Sampling Brownian motion

Initialize the times $t_{1:N}$ and the state r_0.
For n from 0 to $N-1$, iterate the following
- Sample a $\xi_n \sim \text{Normal}(0,1)$.
- Compute $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + \sqrt{2D}(t_{n+1}-t_n)^{1/2}\xi_n$.

2.6 Systems with continuous state-spaces in continuous time

In this section, we focus on modeling the evolution of systems with continuous state-spaces without the simplifying assumption of discrete time.

This is a subtle point as, in the previous section, we had dealt with equations, such as eq. (2.32) or eq. (2.37), for which the coordinates and momenta could be described at every point in time. Yet, we had immediately discretized these equations and obtained discrete evolution equations in time for our variables.

Here, we will not make such approximations. Consequently, we will begin referencing time from now on with t and the system's state with $\boldsymbol{r}(t)$. In this context, we will be using dt to denote an infinitesimally small period of time and $d\boldsymbol{r}(t)$ to denote the corresponding change in the system's state. To help us make this transition to continuous time, we formally introduce *stochastic differential equations* (SDEs) below.

2.6.1 Stochastic differential equations

As we have seen, the transitions of a system in discrete-time may be described by random walks. In such cases, we model each transition by $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + d\boldsymbol{r}_n$. Transition rules like eq. (2.24) remain valid even when the separation between successive time points becomes arbitrarily small. In continuous-time, we need to replace the transition with $\boldsymbol{r}(t+dt) = \boldsymbol{r}(t) + d\boldsymbol{r}(t)$. In the mathematics literature the resulting relations are called SDEs since they describe the evolution of a system with a stochastic component.

If we assume that the stochastic component is time-decorrelated white noise, we can write

$$d\boldsymbol{r}(t) = \underbrace{\mu(t,\boldsymbol{r}(t))dt}_{\text{deterministic}} + \underbrace{\sigma(t,\boldsymbol{r}(t))dW_t}_{\text{stochastic}} \tag{2.56}$$

where $\mu(t,\boldsymbol{r}(t))$ is termed a drift velocity and $\sigma(t,\boldsymbol{r}(t))$ is the noise standard deviation, sometimes also termed *volatility*.

Note 2.23: Stochastic integrals

As we pointed out in note 2.21, the SDE in eq. (2.56) is nothing more than an abbreviation in place of

$$\boldsymbol{r}(t'') - \boldsymbol{r}(t') = \underbrace{\int_{t'}^{t''} dt\, \mu(t,\boldsymbol{r}(t))}_{\text{deterministic integral}} + \underbrace{\int_{t'}^{t''} dW_t\, \sigma(t,\boldsymbol{r}(t))}_{\text{stochastic integral}}$$

and, as can be seen, to fully make sense of this expression, we need to describe the integral $\int_{t'}^{t''} dW_t\, \sigma(t, \boldsymbol{r}(t))$. In the simplest case, where $\sigma(t, \boldsymbol{r}(t))$ is a constant, this integral is equal to the product $\sigma \int_{t'}^{t''} dW_t$, which may be computed from eq. (2.33). A rigorous definition of $\sigma(t, \boldsymbol{r}(t))$ goes beyond the scope of this text.

We see in the note below that for $\boldsymbol{r}(t) = (q(t), p(t))$, the Langevin dynamics given in eq. (2.32) are a special case of eq. (2.56) and the volatility is associated with the strength B of the force fluctuations. Specifically, we show that $\sigma(t, \boldsymbol{r}(t))dW_t$ of eq. (2.56) is equivalent to $dtF_p(t)$ of eq. (2.37). As such, the SDE above is equivalent to

$$\dot{\boldsymbol{r}} = \mu + F(t) \tag{2.57}$$

where $F(t)$ is a stochastic component, described earlier in eq. (2.38), that has units of force for the component of $\boldsymbol{r}(t)$ which coincides with momentum (and for which the force would be subscripted with a p, $F_p(t)$, to suggest a force acting on the momentum).

Note 2.24: Relating the Langevin equation to an SDE

The SDE is a general equation for which $\boldsymbol{r}(t)$ may coincide with position $q(t)$, momentum $p(t)$, both $(q(t), p(t))$, or other states relevant to non-mechanical systems. For sake of concreteness here, we start from a Langevin equation expressed as $\dot{\boldsymbol{r}} = \mu + F(t)$ for which eq. (2.37) is a special case. Integrating both sides, we arrive at

$$\boldsymbol{r}(t+\tau) - \boldsymbol{r}(t) = \int_t^{t+\tau} ds\, \mu + \int_t^{t+\tau} ds\, F(s). \tag{2.58}$$

Similarly, integrating both sides of eq. (2.56), under the assumption of constant $\sigma(t, \boldsymbol{r}(t))$, yields

$$\boldsymbol{r}(t+\tau) - \boldsymbol{r}(t) = \int_t^{t+\tau} ds\, \mu + \sigma \int_t^{t+\tau} dW_t \tag{2.59}$$

which immediately allows us to relate the increments $\int_t^{t+\tau} dsF(s)$ to $\sigma \int_t^{t+\tau} dW_t$. More specifically,

$$\boldsymbol{r}(t) = \begin{bmatrix} q(t) \\ p(t) \end{bmatrix}, \qquad \mu(t, \boldsymbol{r}(t)) = \begin{bmatrix} \frac{p(t)}{m} \\ F(t, \boldsymbol{r}(t)) - \zeta \frac{p(t)}{m} \end{bmatrix}, \qquad \sigma(t, \boldsymbol{r}(t)) = \begin{bmatrix} 0 \\ B \end{bmatrix}. \tag{2.60}$$

from which we recover the underdamped Langevin equation, eq. (2.32), that we saw earlier.

Similarly, reading off from eq. (2.48), for the overdamped Langevin equation we would have

$$\boldsymbol{r}(t) = q(t), \mu(t, \boldsymbol{r}(t)) = F(t, \boldsymbol{r}(t))/\zeta, \sigma(t, \boldsymbol{r}(t)) = B/\zeta. \tag{2.61}$$

Below we show some common examples of SDEs.

Example 2.21: Ornstein-Uhlenbeck process

We now illustrate the versatility of models inspired by Brownian motion. The first is the described overdamped dynamics in potential

$$\zeta dq = -\nabla U(q)dt + BdW_t. \tag{2.62}$$

Of particular interest here is the *Ornstein-Uhlenbeck* (OU) process describing the motion of a point particle evolving according to an overdamped Langevin equation within a harmonic trap, $U(q) = k(q-\mu)^2/2$ where k is the Hookean spring constant. That is

$$\zeta dq = -k(q-\mu)dt + BdW_t. \tag{2.63}$$

The OU process is often used in describing diffusion within a confined environment.

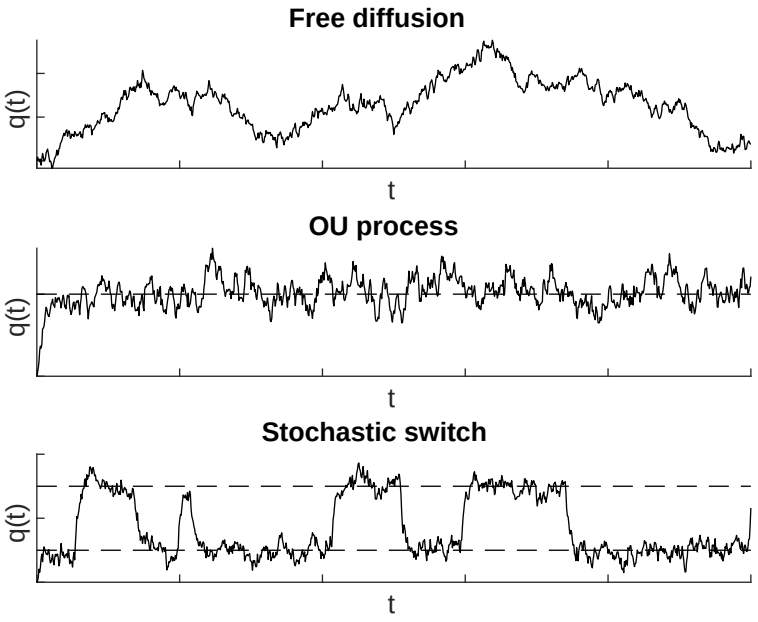


Figure 2.11: Position versus time for free diffusion (top), the OU process (middle) where the particle stays trapped at the potential's mean (μ), and a model describing the stochastic switching between two different harmonic potentials.

Example 2.22: Hop diffusion

The OU process can be further generalized to model a particle hopping between M motion models indexed $\boldsymbol{\sigma}_{1:M}$. Here the hopping can occur in continuous time according to a Markov jump process. One way to model this, following eq. (2.6), is to select the holding states according to

$$S_n|s_{n-1}, \tilde{\pi}_{n-1} \sim \text{Categorical}_{\boldsymbol{\sigma}_{1:M}}(\tilde{\pi}_{n-1}). \tag{2.64}$$

Next, following conventions from eq. (2.7), we then select a holding time for that state

$$H_n|s_n \sim \text{Exponential}\left(\lambda_{s_n}\right). \tag{2.65}$$

The dynamics attained at each time point can then be computed and depend on the state realized

$$\zeta dq = -\nabla U_{s_n}(q)dt + B_{s_n}dW_t \tag{2.66}$$

where $U_{s_n}(q)$ and B_{s_n} set the potential and magnitude of the random force, respectively, in each s_n state.

Example 2.23: Transcription factor binding as an example of hop diffusion

Transcription factors are proteins that modulate gene expression. When tracking these in physical space, we often see transcription factors rapidly diffuse within the cellular nucleus and then reversibly bind to DNA. This binding gives rise to a sudden decrease in their diffusion coefficient. As such, a minimal model describing transcription factor dynamics is to assume that transcription factors switch between two dynamical modes, $M = 2$. Both modes can be assumed to have no potential, $U_{\sigma_1} = U_{\sigma_2} = 0$. The diffusion coefficient in each state of the system, and thus the values for B_{σ_1} and B_{σ_2}, are different.

2.6.2 Fokker-Planck equations

For systems with discrete state-spaces, we have seen in section 2.3 two parallel formulations. One that gives descriptions of how the system passes across its constitutive states and allowed us to sample individual trajectories. The other allows for probabilities over states to quantify the dynamics of the system globally over the entire state-space. The first formulations uses sampling equations that can readily implemented computationally (*i.e.* the Gillespie algorithm); while, the second formulation uses master equations that typically need to be solved analytically or numerically.

For systems described in continuous state-space and time a similar picture also applies. SDEs describe the evolution of the system's state $\boldsymbol{R}(t)$ through time and are analogous to the transition rules. However, as the system changes stochastically, $\boldsymbol{R}(t)$ is not determined uniquely but only probabilistically. As such, we can speak of a probability density of observing a realization r of $\boldsymbol{R}(t)$ at a specific time t.

That is, just as we did in deriving master equations, we may write an initialization rule for R_0 and also write a probability of $\boldsymbol{r}(t)$ being inside some volume V as $\int_V d\boldsymbol{r}\, p(\boldsymbol{r},t)$ where the volume element is understood as $d\boldsymbol{r}$.

This density, which is an function of both space and time and is written as $p(\boldsymbol{r},t)$, satisfies a *Fokker-Planck equation*. The Fokker-Planck is analogous to the master equation.

Our goal is now to derive the Fokker-Planck. As we did in deriving master equations in eq. (2.8), to derive a Fokker-Planck equation, we begin with an argument for how probability changes. To do so, we focus on a volume V in the system's state-space and its boundary which we denote with S. For mechanical systems in 1 dimension, $d\boldsymbol{r}$ takes the explicit form $dq\, dp$.

To derive the Fokker-Planck, we write that the total amount of probability density flowing out of S as

$$\int_S dS \cdot \dot{\boldsymbol{r}} p\left(\boldsymbol{r},t\right) \tag{2.67}$$

where the normal to the surface is pointed outward, by convention, and the velocity $\dot{\boldsymbol{r}}$ is given by the SDE of eq. (2.56). For $dS \cdot \dot{\boldsymbol{r}}$ positive, probability flows out of the volume, V. The decrease in time of the probability enclosed in the volume, V, can also be written as

$$-\frac{\partial}{\partial t}\int_V d\boldsymbol{r} p\left(\boldsymbol{r},t\right). \tag{2.68}$$

Equating eq. (2.68) and eq. (2.67) and invoking the divergence theorem, we obtain

$$-\frac{\partial}{\partial t}\int_V d\boldsymbol{r} p\left(\boldsymbol{r},t\right) = \int_V d\boldsymbol{r} \nabla \cdot \left(\dot{\boldsymbol{r}} p\left(\boldsymbol{r},t\right)\right). \tag{2.69}$$

Since eq. (2.69) holds for any volume V, the *continuity equation* follows

$$\frac{\partial}{\partial t} p\left(\boldsymbol{r},t\right) + \nabla \cdot \left(\dot{\boldsymbol{r}} p\left(\boldsymbol{r},t\right)\right) = 0. \tag{2.70}$$

Now, inserting the SDE $\dot{\boldsymbol{r}} = \mu + F$ from eq. (2.57) into eq. (2.70) yields

$$\frac{\partial}{\partial t} p\left(\boldsymbol{r},t\right) + \nabla \cdot \left(\left(\mu + F\right) p\left(\boldsymbol{r},t\right)\right) = 0. \tag{2.71}$$

From now on, our goal is to eliminate the fluctuating force from the expression above. The end result, for which a derivation is provided in the box below, reads

$$\frac{\partial}{\partial t} p\left(\boldsymbol{r},t\right) + \nabla \cdot \left(\mu p\left(\boldsymbol{r},t\right)\right) = +\frac{1}{\zeta^2} \nabla \cdot B\left(\nabla \cdot B p\left(\boldsymbol{r},t\right)\right). \tag{2.72}$$

Note 2.25: Eliminating the fluctuating force from the Fokker-Planck equation

Here we derive the Fokker-Planck equation for the probability, $p(\boldsymbol{r},t)$, for r given by the SDE in eq. (2.56). We start with eq. (2.71) that we re-write as

$$\frac{\partial}{\partial t}p(\boldsymbol{r},t)+p(\boldsymbol{r},t)\,\nabla\cdot\mu=-\mu\cdot\nabla p(\boldsymbol{r},t)-\nabla\cdot(F(t)p(\boldsymbol{r},t))\,. \tag{2.73}$$

Using the integrating factor method, in a way very similar to what we did in example 2.18, we arrive at

$$p(\boldsymbol{r},t)=p(\boldsymbol{r}_0)\,e^{-t\nabla\cdot\mu}-\int_0^t ds e^{-(t-s)\nabla\cdot\mu}\mu\cdot\nabla p(\boldsymbol{r},s) \tag{2.74}$$

$$-\int_0^t ds e^{-(t-s)\nabla\cdot\mu}\nabla\cdot(F(s)p(\boldsymbol{r},s))\,. \tag{2.75}$$

Now eq. (2.74) can be inserted into the second term on the right hand side of eq. (2.73) to yield

$$\frac{\partial}{\partial t}p(\boldsymbol{r},t)+\nabla\cdot(\mu p(\boldsymbol{r},t))=$$
$$-\nabla\cdot\left(F(t)\left(p(\boldsymbol{r}_0)\,e^{-t\nabla\cdot\mu}-\int_0^t ds e^{-(t-s)\nabla\cdot\mu}\mu\cdot\nabla p(\boldsymbol{r},s)-\int_0^t ds e^{-(t-s)\nabla\cdot\mu}\nabla\cdot(F(s)p(\boldsymbol{r},s))\right)\right)\,. \tag{2.76}$$

Averaging both sides of eq. (2.76) with respect to noise and invoking eq. (2.61) (for the overdamped Langevin equation) yields

$$\frac{\partial}{\partial t}p(\boldsymbol{r},t)+\nabla\cdot(\mu p(\boldsymbol{r},t))=+\frac{1}{\zeta^2}\nabla\cdot B(\nabla\cdot Bp(\boldsymbol{r},t))\,. \tag{2.77}$$

The exercise can be repeated for underdamped Langevin equation using eq. (2.60).

A few remarks on noise averaging are warranted here.

First, the noise average of the first term on the right hand side of eq. (2.76) is zero. This is because the initial condition, $p(\boldsymbol{r}_0)$, has no stochastic component and the average of $F(t)$ is zero.

Second, from eq. (2.74) it is clear that $p(\boldsymbol{r},t)$ itself has a stochastic component. As such we could re-insert eq. (2.74) into the second term on the right hand side of eq. (2.73). This would yield a series expansion termed the *Kramers-Moyall expansion*. By ignoring higher order terms in the Kramers-Moyall expansion, we are assuming that, up to second order in the fluctuating force, the probability appearing in the last term of eq. (2.76) is independent of the fluctuating force. As such we only compute the covariance of the force as per eq. (2.46), not, say, $\langle F(t)F(s)F(s')F(s'')\rangle$.

One of the simplest forms of eq. (2.72) is the case where $\mu=0$ and B is constant. This coincides to a special case of the SDE provided in eq. (2.56) which corresponds to Brownian motion. For the special case where every component of B is constant and identical, we speak of *isotropic free diffusion* and the resulting Fokker-Planck equation, sometimes called the *diffusion equation*, reads as follows

$$\frac{\partial}{\partial t}p(\boldsymbol{r},t)=D\nabla^2 p(\boldsymbol{r},t)\,. \tag{2.78}$$

For B components constant but different along different directions, we would speak of an *anisotropic diffusion equation*. Similarly, for $\mu\neq 0$, we speak of a *diffusion equation with drift*. The derivation of eq. (2.76) is provided in the note below.

Note 2.26: Derivation of the diffusion equation

We start from the special case eq. (2.56) where every component of B is constant and identical and $\mu=0$ to derive eq. (2.76). Under these assumptions, eq. (2.56) immediately simplifies to

$$d\boldsymbol{r}(t)=BdW_t \tag{2.79}$$

where, for the case of 3D diffusion, eq. (2.79) is understood as 3 separate equations, one for each independent direction in Cartesian coordinates. From eq. (2.79), the general Fokker-Planck of eq. (2.76) immediately reduces to eq. (2.76).

With open boundary conditions (probability decays to zero at infinity) and initial conditions reflecting perfect certainty as to the location $\boldsymbol{r}(t_{init})$ of the diffusing particle, $p\left(\boldsymbol{r}(t_{init})\right) = \delta_{\boldsymbol{r}_0}(\boldsymbol{r}(t_{init}))$, the 3-dimensional diffusion equation can be solved and the solution reads

$$p(\boldsymbol{r},t) = \frac{1}{\left(4\pi D(t-t_{init})\right)^{3/2}} e^{-\frac{(\boldsymbol{r}-\boldsymbol{r}(t_{init}))^2}{4D(t-t_{init})}}. \tag{2.80}$$

This solution describes transition probability rules that we have already written in explicit form in eq. (2.55).

2.6.3 A case study in thermal physics*

It is of interest to the Physical Sciences to ask under what circumstance we can derive the diffusion equation for systems with position and momentum degrees of freedom. As we will need to discriminate between momentum and the probability density ($p(\boldsymbol{r},t)$), we will use p for momentum and $p(\boldsymbol{r},t)$ for probability density.

We begin by considering the overdamped Langevin equation, eq. (2.62). Following the recipe in note 2.25, we arrive at the following Fokker-Planck equation

$$\frac{\partial}{\partial t}p(\boldsymbol{r},t) - \nabla\cdot\left(\nabla U(q)p(\boldsymbol{r},t)\right) = D\nabla^2 p(\boldsymbol{r},t). \tag{2.81}$$

which is sometimes called the *Smoluchowski equation*.

Another Fokker-Planck equation can be obtained by considering the underdamped Langevin equation, eq. (2.32). Again, following a recipe similar to note 2.25, we arrive at a different Smoluchowski equation

$$\frac{\partial}{\partial t}p(\boldsymbol{r},t) + \frac{q}{m}\cdot\nabla p(\boldsymbol{r},t) + \nabla_p\cdot\left(\left(-\nabla U(q) - \frac{\zeta}{m}p\right)p(\boldsymbol{r},t)\right) = D\zeta^2\nabla_p^2 p(\boldsymbol{r},t) \tag{2.82}$$

where, to be clear, we have subscripted the ∇ by momentum, ∇_p, if it is a gradient with respect to momentum. Otherwise, the ∇ is understood as a gradient with respect to location, $\partial/\partial q$.

It is clear that for a mechanical system where $p = m\dot{q}$, we should be able to re-parametrize the underdamped Smoluchowski equation from eq. (2.82) into a partial differential equation that only depends on position and time. Indeed, the end result, derived in the note below, reads

$$\frac{\partial p(\boldsymbol{r},t)}{\partial t} = \frac{1}{\zeta}\nabla_p\cdot\left(\nabla U(q)p(\boldsymbol{r},t)\right) + \frac{k_BT}{\zeta}\nabla^2 p(\boldsymbol{r},t) - \frac{m}{\zeta}\frac{\partial^2}{\partial t^2}p(\boldsymbol{r},t) \tag{2.83}$$

where the second time derivative of the probability suggests partial time-reversibility on timescales of m/ζ.

Note 2.27: The underdamped Fokker-Planck equation

To derive the Fokker-Planck equation coinciding with the underdamped Langevin equation for mechanical systems, eq. (2.40), we start by inserting dynamics for $\dot{q}$ and $\dot{p}$, eqs. (2.36) and (2.37), into the continuity equation, eq. (2.70),

$$\frac{\partial}{\partial t}p(\boldsymbol{r},t) + \nabla\cdot\left(\dot{q}p(\boldsymbol{r},t)\right) + \nabla_p\cdot\left(\dot{p}p(\boldsymbol{r},t)\right) = 0 \tag{2.84}$$

which immediately yields

$$\frac{\partial}{\partial t}p(\boldsymbol{r},t) + \frac{p}{m}\cdot\nabla p(\boldsymbol{r},t) + \nabla_p\cdot\left(\left(-\nabla U(q) - \frac{\zeta}{m}p + F_p(t)\right)p(\boldsymbol{r},t)\right) = 0 \tag{2.85}$$

*This is an advanced topic and could be skipped on a first reading.

where we subscript the stochastic force, $F_p(t)$, with p to emphasize that this force is acting on the momenta.

As a first step toward capturing the statistics of $F_p(t)$, eq. (2.39), into the partial differential equation, we first average over all values of the momentum

$$\frac{\partial}{\partial t}\left(\int dpp(\boldsymbol{r},t)\right)+\int dp\left(\frac{p}{m}\cdot\nabla p(\boldsymbol{r},t)\right)+0=0 \tag{2.86}$$

where the third term on the left hand side is zero as $p(\boldsymbol{r},t)$ vanishes when evaluated for infinite values of the momentum, p. We can re-write the above as

$$\frac{\partial}{\partial t}\bar{p}(\boldsymbol{r},t)+\frac{1}{m}\nabla\cdot\langle p\rangle=0 \tag{2.87}$$

where $\bar{p}$ is the momentum averaged transition density. In other words, we say

$$\bar{p}(\boldsymbol{r},t)=\int dpp(\boldsymbol{r},t)=p(q,t). \tag{2.88}$$

Another note is in order regarding eq. (2.87). Certainly, at equilibrium, $\langle p\rangle=0$. However on timescales shorter than or on order of m/ζ, the density p may strongly depend on initial conditions for which the average momentum may be non-zero.

Now, if we could express $\langle p\rangle$ in terms of position we would be done as we would have functions of position on both sides of eq. (2.87). So to find $\langle p\rangle$, we need an equation of motion for $\langle p\rangle$.

To do this, we note that the random forces acting on the momentum are Gaussian. For this reason, we multiple both sides of eq. (2.85) by momentum and integrate over momentum,

$$\int dp\left(p\frac{\partial}{\partial t}p(\boldsymbol{r},t)+\frac{p}{m}p\cdot\nabla p(\boldsymbol{r},t)+p\nabla_p\cdot\left(\left(-\nabla U(q)-\frac{\zeta}{m}p+F_p(t)\right)p(\boldsymbol{r},t)\right)\right)=0 \tag{2.89}$$

which we re-write as

$$\frac{\partial}{\partial t}\langle p\rangle+\int dp\frac{p}{m}p\cdot\nabla p(\boldsymbol{r},t)-\int dp\left(\left(-\nabla U(q)-\frac{\zeta}{m}p+F_p(t)\right)p(\boldsymbol{r},t)\right)=0. \tag{2.90}$$

The third term was simplified through integration by parts. Taking the divergence of both sides of the above with respect to q gives

$$\frac{\partial}{\partial t}\nabla\cdot\langle p\rangle+\frac{\mathbf{1}}{m}\int dp\,(p\cdot\nabla)^2\,p(\boldsymbol{r},t)-\nabla\cdot\int dp\left(\left(-\nabla U(q)-\frac{\zeta}{m}p+F_p(t)\right)p(\boldsymbol{r},t)\right)=0. \tag{2.91}$$

Assuming momenta in all directions are decoupled and averaging over noise yields

$$\frac{\partial}{\partial t}\nabla\cdot\langle p\rangle+\frac{\mathbf{1}}{3m}\nabla^2\langle p^2\rangle-\nabla\cdot(-\nabla U(q)\bar{p}(\boldsymbol{r},t))+\frac{\zeta}{m}\nabla\cdot\langle p\rangle=0 \tag{2.92}$$

where in obtaining the second term on the right hand side we wrote,

$$\begin{aligned}&\int dp\,(p\cdot\nabla)^2\,p(\boldsymbol{r},t)\\&=\partial_x^2\langle p_x^2\rangle+\partial_y^2\langle p_y^2\rangle+\partial_z^2\langle p_z^2\rangle\\&=\nabla^2\frac{1}{3}\langle p^2\rangle\end{aligned} \tag{2.93}$$

under the assumption that the average over momenta in any direction is the same (*e.g.* $\partial_x^2\langle p_y^2\rangle=\partial_x^2\langle p_z^2\rangle$). Inserting eq. (2.87) into eq. (2.92) we recover

$$-m\frac{\partial^2}{\partial t^2}\bar{p}(\boldsymbol{r},t)+\frac{\mathbf{1}}{3m}\nabla^2\langle p^2\rangle-\nabla\cdot(-\nabla U(q)\bar{p}(\boldsymbol{r},t))-\zeta\frac{\partial}{\partial t}\bar{p}(\boldsymbol{r},t)=0. \tag{2.94}$$

The above is still not a closed set of equations as it depends on $\langle p^2 \rangle$. The next logical step would be to evaluate the evolution equation for $\langle p^2 \rangle$ just as we had before for the first moment.

Instead what we do is to assume a moment closure relation. That is, we assume that the mean square momentum is dictated by the temperature. In other words, we make the equilibrium assumption that $p(\boldsymbol{r}, t)$ is separable in position and momenta, $p(q, t)p(p, t)$,

$$\begin{aligned} \frac{\mathbf{1}}{3m}\nabla^2\langle p^2 \rangle &= \frac{\mathbf{1}}{3m}\nabla^2 \int dp p(\boldsymbol{r}, t)p^2 \\ &= \frac{\mathbf{1}}{3m}\nabla^2 p(q) \int dp p(p, t)p^2 \\ &= 3mk_BT\frac{\mathbf{1}}{3m}\nabla^2\bar{p}(\boldsymbol{r}, t). \end{aligned} \tag{2.95}$$

Thus, we finally have

$$-m\frac{\partial^2}{\partial t^2}\bar{p}(\boldsymbol{r}, t) + k_BT\nabla^2\bar{p}(\boldsymbol{r}, t) - \nabla \cdot (-\nabla U(q)\bar{p}) - \zeta\frac{\partial}{\partial t}\bar{p} = 0. \tag{2.96}$$

Re-writing in more convenient form, we have

$$\frac{\partial\bar{p}(\boldsymbol{r}, t)}{\partial t} = \frac{1}{\zeta}\nabla \cdot (\nabla U(q)\bar{p}(\boldsymbol{r}, t)) + \frac{k_BT}{\zeta}\nabla^2\bar{p}(\boldsymbol{r}, t) - \frac{m}{\zeta}\frac{\partial^2}{\partial t^2}\bar{p}(\boldsymbol{r}, t) \tag{2.97}$$

where we interpret k_BT/ζ as the diffusion coefficient. Eq. (2.97) is a Smoluchowski equation describing underdamped dynamics.

We can ask, where has the effect of the random force gone since it seemed to have vanished by integration or averaging? It has set the magnitude of the expectation of $\langle p^2 \rangle$.

The Fokker-Planck equation given by eq. (2.97) is more general than the overdamped form, eq. (2.76), typically encountered when m/ζ is fast as compared to all timescales present in the problem (in particular, say, the inverse frequency of a potential minimum trapping the point mass).

2.7 Exercise problems

Exercise 2.1: Poisson master equation

Show that the solution to the master equation for the birth process is the Poisson distribution.

Exercise 2.2: Poisson convolution

Show that the convolution of multiple Poisson distributions remains a Poisson distribution.

Exercise 2.3: Birth-death

RNA molecules are produced independently from a single DNA molecule. RNA molecules are also degraded spontaneously. (i) Simulate this birth-death process for RNA (assuming a birth rate greater than the death rate). (ii) Plot the amount of RNA as a function of time until the system reaches steady state. (iii) Explain how this steady state is related to the production and degradation rates.

Exercise 2.4: Numerical integrators

Consider a 1D harmonic oscillator whose equations of motion are

$$\frac{dq(t)}{dt} = \frac{p(t)}{m}, \qquad \frac{dp(t)}{dt} = -kq(t).$$

Assume the system is initialized at $q(t_0) = q_0$ and $p(t_0) = p_0$.

1. Derive the general solution for $\boldsymbol{r}(t) = (q(t), p(t))$.
2. Derive the transition rules for the oscillator's trajectory $\boldsymbol{r}_n = (q_n, p_n)$ at times $t_n = t_0 + n\tau$ with the three approximations mentioned in note 2.22. Specifically, use
   - $F(t, q(t)) \approx F(t_n, q(t_n))$
   - $F(t, q(t)) \approx F(t_{n+1}, q(t_{n+1}))$
   - $F(t, q(t)) \approx \frac{1}{2}\left(F(t_n, q(t_n)) + F(t_{n+1}, q(t_{n+1}))\right)$
3. Use $m = 1$, $k = 1$, $q_0 = 0$, $p_0 = 1$ and $\tau = 0.25$ to compute specific trajectories and compare with the exact solution.
4. Compute the energy along the three trajectories computed above and compare with the energy along the exact trajectory.

Exercise 2.5: Stochastic binary decisions

Cell fate decisions are often based on small initial fluctuations that are amplified and reinforced (through feedback) over time. These events are called *stochastic binary decisions* (Artyomov et al., PNAS, 104, 18958, 2007). We will now simulate such a process. Consider the following set of chemical reactions:

$$A_1 \xrightarrow{k_1} E + A_1$$

$$A_2 \xrightarrow{k_2} S + A_2$$

$$E + A_1 \xrightarrow{k_3} E + A_1^*$$

$$A_1^* \xrightarrow{k_4} E + A_1^*$$

$$S + A_1 \xrightarrow{k_5} S + A_1^{INACTIV}$$

$$S \xrightarrow{k_d} \emptyset$$

$$E \xrightarrow{k_d} \emptyset$$

where: A_1 is an agonist and A_2 is its antagonist; E is an enzyme that converts A_1 into its protected form A_1^*; and A_1^*, in turn, stimulates the production of E (positive feedback). We are interested in the steady state amount of A_1^*. If S is present it can permanently de-activate A_1.

1. Start with 10 agonists and 10 antagonists with $k_1 = k_2 = k_d = k_4 = 1, k_3 = 100, k_5 = 100$. Simulate the process to completion using the Gillespie algorithm and histogram the final amount of A_1^*. Explain, the result you obtain.
2. Repeat the simulation and histograming starting with 1000 agonists and 1000 antagonists. Explain, how your histogram differs from the previous one.
3. If you had solved the corresponding rate equations explain what you would expect the steady state population of A_1^* to look like.

Exercise 2.6: The genetic toggle switch and stochastic bistability

The *toggle switch* (Gardner et al. Nat. 403, 339, 2000) is a common feedback loop motif in Systems Biology and it exhibits a behavior called "stochastic bistability". We will now simulate this behavior. Consider the following

chemical reactions involving two proteins, A and B:

$$A \xrightarrow{k_d} \emptyset$$

$$B \xrightarrow{k_d} \emptyset$$

$$g_A \xrightarrow{k_p} g_A + A$$

$$g_B \xrightarrow{k_p} g_B + B$$

$$g_A + B \underset{k_r}{\overset{k_f}{\leftrightarrow}} g_A^*$$

$$g_B + A \underset{k_r}{\overset{k_f}{\leftrightarrow}} g_B^*$$

where: k_d are degradation rates and k_p are production rates for both proteins; g_A is the gene responsible for the production of A which can converted into an inactive form g_A^* by binding to B and *vice versa* for g_B. Assume you only have one gene available in the cell so that $g_A + g_A^* = 1$ and $g_B + g_B^* = 1$. Also, assume throughout that $g_A^* + g_B^* = 1$, $k_d < k_p$ and $k_r < k_f n_B, k_f n_A$.

1. Simulate the chemical reactions starting with $n_A = 0$ and $n_B = 0$ for multiple time steps. Adjust your rates until you achieve stochastic switching events between phases where A exceeds B in number and B exceeds A. You should see stochastic hopping between two solutions (which we call "fixed points").
2. Would you expect to see this stochastic switching occur if you had started with a large amount of n_A and n_B initially? In technical language, qualitatively explain how the fixed point structure changes for the corresponding rate equations.
3. The condition that $g_A^* + g_B^* = 1$ is called the *exclusive switch*. Relax this condition and re-simulate the toggle switch. What new fixed point appears?

Exercise 2.7: Third order reactions

Consider the following reactions: $A + A \xrightarrow{\lambda_1} A_2$ followed by $A_2 + A \xrightarrow{\lambda_2} A_3$. Show that when λ_1 vastly exceeds λ_2, *i.e.* when $\lambda_1 \gg \lambda_2$, then the rate of production of A_3 predicted from mass action is the same as that expected from the effective reaction $A + A + A \xrightarrow{\lambda_3} A_3$.

Exercise 2.8: Fluctuation-Dissipation theorem

Start with the following Langevin equation, where rather than a scalar friction coefficient, we now have what is termed a frequency dependent memory kernel

$$m\ddot{q} = -\int_0^t dt' \zeta(t - t')\dot{q}(t') + F_p(t).$$

Show that the function, $f(t)$, for which the fluctuation-dissipation theorem, $\langle F_p(t) \cdot F_p(t') \rangle = f(t - t')$, is satisfied is $f(t) = 3k_B T \zeta(t)$.

Chapter 3

Likelihoods

By the end of this chapter, we will have presented

- *How to manipulate likelihoods*
- *The principle of maximum likelihood*
- *Noise models in likelihoods*
- *The expectation-maximization algorithm*

3.1 Quantifying measurements with likelihoods

The models we formulate in order to analyze data contain mathematical *variables* whose values are unknown and others whose values are known. Measurements, which we typically denote with w or $w_{1:N}$, are examples of variables with known values. In real-life applications, such values are specified in the experiments we seek to analyze.

In a model, variables with unknown values are of two kinds: those that we care to estimate, which we call *parameters* and often denote with θ or $\theta_{1:K}$; and those that we may not care to estimate, which we term *latent* or *nuisance* variables, that we explore later in section 3.3.

The distinction between parameters and latent variables is generally not mathematically clear cut. After all, all are represented with random variables whose statistics are interrelated according to rules we develop in the first chapter. Nevertheless, from the modeling perspective, it is always helpful to have a good idea of what variables need to be estimated and what variables are needed only because they facilitate the model description.

Example 3.1: Example of latent variables

Consider particles trapped in a harmonic potential, *e.g.* whose trajectory is described by a Ornstein-Uhlenbeck process. We may only care to learn the harmonic force constant trapping particles and the particle's friction coefficients from grainy observations, not the actual particle trajectories themselves. In this case, the harmonic force constant and friction coefficient constitute the parameters we care to learn and the particle positions are latent variables.

We have previously seen parameters inside generative models of dynamical systems. For example, in discrete-space discrete-time systems, section 2.4, each constitutive state is sampled from a Categorical distribution whose parameters include the probabilities of state-to-state transitions; while, in discrete-space continuous-time systems, section 2.3, each holding state is sampled from a Categorical distribution whose parameters include the transition probabilities and each holding period is sampled from an Exponential distribution whose parameters include the holding rate. The distributions forming a generative model capture our *model's structure* and have parameter, θ, dependencies.

From the model structure and observations, $w_{1:N}$, the goal may therefore be to parameter estimates, $\hat{\theta}$. In this chapter, we do so following what is often termed the *frequentist paradigm*. In chapter 4, we will discuss how

to obtain full distributions over θ, not just estimates $\hat{\theta}$. We will do so within a *Bayesian paradigm*.

Both strategies, whether frequentist or Bayesian, employ *likelihoods*. As formulating likelihoods given model structures and observations is the starting point of frequentist inference, the focus of this chapter is in describing likelihoods and using them in frequentist inference.

3.2 Estimating parameters with maximum likelihood

Within the frequentist paradigm, parameters are variables whose values are to be learned from the data, $w_{1:N}$. To estimate parameters, we ask:

- What is the likelihood of having observed the sequence $w_{1:N}$ under the assumptions of our model structure (dictating the mathematical form of the likelihood)?
- Armed with this likelihood evaluated at $w_{1:N}$, we now ask: What values for θ maximize this likelihood?

Taken together, these two concepts form the *maximum likelihood principle* that we illustrate with a simple example below.

Example 3.2: Estimating diffusion coefficients using maximum likelihood

We consider the position, $r(t)$, of a normally diffusing particle in one dimension for now (generalization to higher dimensions is straightforward).

We assume that the position is assessed at discrete and equally spaced, $t_n = t_0 + n\tau$, time levels such that, at the n^{th} time level, we write $w_n = r(t_n) = r_n$.

From eq. (2.80), we immediately write

$$p(r_n|r_{n-1}, D) = \frac{1}{(4\pi D\tau)^{1/2}} e^{-\frac{(r_n - r_{n-1})^2}{4D\tau}} \tag{3.1}$$

where the value of the diffusion coefficient D, the only unknown, remains to be determined. The likelihood of observing the full trajectory, $r_{1:N}$, is

$$p(r_{1:N}|D) = p(r_N|r_{N-1}, D) \cdots p(r_3|r_2, D)p(r_2|r_1, D)p(r_1|D). \tag{3.2}$$

As the initial position of the particle, r_1, is independent of D, we write $p(r_1|D)$ as $p(r_1)$. By inserting eq. (3.1) into eq. (3.2), we arrive at

$$p(r_{1:N}|D) = \frac{1}{(4\pi D\tau)^{(N-1)/2}} e^{-\sum_{n=2}^{N} \frac{(r_n - r_{n-1})^2}{4D\tau}} p(r_1) \tag{3.3}$$

whose logarithm, termed the *log likelihood* reads

$$\log p(r_{1:N}|D) = -\frac{(N-1)}{2} \log (4\pi D\tau) - \sum_{n=2}^{N} \frac{(r_n - r_{n-1})^2}{4D\tau} + \log p(r_1). \tag{3.4}$$

As the logarithm is monotonic with the original likelihood, we often choose to maximize the logarithm over the function itself.

Now maximizing the log likelihood by setting to zero its derivative with respect to D yields

$$-\frac{(N-1)}{2\hat{D}} + \sum_{n=2}^{N} \frac{(r_n - r_{n-1})^2}{4\hat{D}^2\tau} = 0 \tag{3.5}$$

where $\hat{D}$ is the estimated value of D maximizing the likelihood. Re-arranging the above, we find

$$6\hat{D} = \frac{1}{N-1} \sum_{n=2}^{N} \frac{(r_n - r_{n-1})^2}{\tau} \tag{3.6}$$

which we had previously seen in eq. (2.49).

As we see in the example, as well as throughout the text, we often maximize a likelihood's logarithm by contrast to maximizing the likelihood directly as both the original function as well as its logarithm have identical maxima.

Historically, maximizing logarithms may have arisen as this is a mathematically simpler function to maximize by virtue of commonly assumed likelihoods (for the same reason that it is often simpler to deal with free energy rather than partition functions in thermal physics).

Today, maximizing log likelihoods is a question of practical relevance. Numerically, the log likelihood is essential in avoiding numerical underflow as likelihoods over long data sequences, that is $p(w_{1:N}|\theta_{1:K})$ for appreciable N, can become very small indeed.

This example highlights why the logarithm of the likelihood rather than the likelihood itself is often maximized. Another important point that we draw from this example is that the likelihood captures *all of the physics* of the underlying process.

Although physically accurate, the example above has one critical limitation that renders it inappropriate to analyze measurements obtained experimentally. Namely, it assumes that positions are measured with no uncertainty. In particular, we assumed no discrepancy between, say, the measurement of a particle's position, r_n and the observation, w_n. We now lift this assumption as it is clear that measurement always introduces error. That is, there exists a probabilistic relationship between the variables we care about, say positions $r_{1:N}$, and observations, $w_{1:N}$.

3.3 Observations and the associated measurement noise

In the previous chapter, we presented a number of broad dynamical forward models. The origin of uncertainty and the reason we formulated them probabilistically stems from the stochastic dynamics inherent in such systems. For example, in the case of the Langevin equation, **??**, and ensuing Fokker-Planck equation, **??**, the source of uncertainty originates from thermal agitation of the particle's surrounding medium. While thermal fluctuations and temperature are not made as explicit in the context of Gillespie's algorithm, **??**, the same is true, fundamentally, of the Gillespie simulation, **??**, where thermal fluctuations set which reaction occurs next.

Yet, even if the dynamics of our systems were deterministic, we would still require a (probabilistic) likelihood. The reason for this is simple: we have *measurement noise*. Accounting for the uncertainty inherent to the measurement process is a requirement in a full description of a likelihood and introduces the concept of *latent variables* which we denote by $r_{1:N}$. In full generality, these coincide with the hidden states of a system indirectly monitored through measurement, $w_{1:N}$.

We focused on dynamics here but the same is true of iid processes. The latent variables realized by an iid process may be deterministic (*i.e.* those generated from a Bernoulli distribution with parameter $\pi = 1$) but measurement noise may obscure the realization of the latent variable.

This discussion immediately forces us to separate the objects that we generally referred to as parameters $\theta_{1:K}$ into two distinct categories. The first are the parameters of the *observation distribution* (sometimes called an emission distribution in the machine learning literature), characterizing the measurement noise, which we label $\phi_{1:I}$. These parameters are often called *observation parameters* (or, equivalently, emission parameters). The other parameters are parameters of the generative model not involved in the observation distribution (that we previously called $\theta_{1:K}$) but that we now call $\psi_{1:J}$. Together $\theta = \{\phi, \psi\}$.

In general, the parameters of the observation distribution depend on the state of the latent variables and, in the most general case, we write $\phi(r_{1:N})$ for those parameters associated to the n^{th} time level (we discount the even greater generality in which the observation distribution depends on time).

Causality itself already helps us simplify this dependency as we write

$$w_n|\phi(r_{1:n}) \sim \mathbb{F}(\phi(r_{1:n})) \tag{3.7}$$

where $\mathbb{F}(\phi(r_{1:n}))$ are parameters of the measurement distribution, $\mathbb{F}$, from which our observations are drawn. Causality is manifest in the fact that the observation parameters are independent of $r_{n+1:N}$.

In the most general case, the unknowns now include: $r_{1:N}$ and $\theta = \{\phi, \psi\}$.

Less generally, the measurement apparatus may have been pre-calibrated and, in this case, the ϕ are known. The unknowns are then reduced to $r_{1:N}$ and $\theta = \{\psi\}$.

An important simplifying assumption about the observation model contained in eq. (3.7) is often to say that

$$w_n|\phi(r_n) \sim \mathbb{F}(\phi(r_n)). \tag{3.8}$$

That is, the current measurement only depends on the current observation. We note that the measurement model is independent of whether the r_n is iid or evolves according to a Markov or more complex dynamical model.

Note 3.1: Generative models with measurement uncertainty

For iid random variables, a generative model accounting for measurement uncertainty reads

$$\begin{aligned} r_n|\theta &\sim \mathbb{P}(\theta) \\ w_n|\phi(r_n) &\sim \mathbb{F}(\phi(r_n)) \end{aligned} \tag{3.9}$$

where the second line includes the measurement model under the assumption of eq. (3.8). Similarly, for Markov random variables under the assumption of eq. (3.8), we have

$$\begin{aligned} r_n|\theta(r_{n-1}) &\sim \mathbb{P}(\theta(r_{n-1})) \\ w_n|\phi(r_n) &\sim \mathbb{F}(\phi(r_n)) \end{aligned} \tag{3.10}$$

where, under the Markov assumption, we have made explicit the dependency of θ on the previous realization of the system, r_{n-1}.

To illustrate a concrete example of a generative model for R_n as well as an associated observation model, below we give an example of a freely diffusing particle with a Normal observation distribution.

Note 3.2: Gaussian observation distribution models

Here we consider a freely diffusing particle whose transition probability density is described by

$$r_n|r_{n-1}, B^2 \sim \text{Normal}(r_{n-1}, B^2). \tag{3.11}$$

Here r_n plays the role of a latent variable and the observation is, as usual, denoted by w_n. Following the assumption made in eq. (2.38) and assuming a Normal observation model, we write the full forward model as follows

$$r_n|r_{n-1}, B^2 \sim \text{Normal}(r_{n-1}, B^2) \tag{3.12}$$

$$w_n|r_n, \sigma^2 \sim \text{Normal}(r_n, \sigma^2) \tag{3.13}$$

where σ^2 is the variance of the observation distribution. For completeness, we would say $\phi(r_n) = \{r_n, \sigma^2\}$.

So far, we have only discussed observation models whose output, w_n, depends on the instantaneous realization of r_n. However, if required by the measurement process, observation models can accommodate more complex scenarios.

Note 3.3: More complex observation models

Here we envision snapshots of a particle's position tracked in two dimensions by a camera system. For simplicity, we imagine that the measurement w_n, obtained at time t_n, is the result of integration over all positions of the particle attained over the exposure time window, $\tau = t_n - t_{n-1}$. Thus the measurement model here reads

$$w_n|\bar{r}_n(t), \sigma^2 \sim \text{Normal}\left(\bar{r}_n(t) = \frac{1}{t_n - t_{n-1}} \int_{t_{n-1}}^{t_n} dt r(t), \sigma^2\right) \tag{3.14}$$

where $\bar{r}_n(t)$ denotes the integrated value of the position over τ ending at t_n.

Above are two examples of continuous space models. Below we include a note on discrete state models which follows similar logic.

Note 3.4: Hidden Markov models

Here is an example of a full forward model for a discrete state system with a measurement model under the assumption of eq. (2.38)

$$S_n|s_{n-1} = \sigma_m, \boldsymbol{\pi}_{n-1} \sim \text{Categorical}_{\sigma_{1:M}}(\tilde{\pi}_{\sigma_m}) \tag{3.15}$$

$$W_n|\phi(s_n) \sim F(\phi(s_n)). \tag{3.16}$$

The model above is of fundamental importance and forms the basis of a model often understood as a paradigm of data analysis: the *hidden Markov model* (HMM). We turn to HMMs in much greater depth in chapter 8. Typically, in the HMM, it is commonplace to re-write $\phi(s_n)$ as ϕ_{s_n}.

3.4 Variants of a likelihood

3.4.1 Completed likelihoods

When we introduce measurement noise, and the associated concept of latent variables, likelihoods often become more complicated.

In this case, as we will see through examples, it is often simpler to compute the joint likelihood over the observations and latent variables $p(w_{1:N}, r_{1:N}|\theta)$ than it is to perform a marginalization over the latent variables and compute $p(w_{1:N}, r_{1:N}|\theta)$.

These joint likelihoods are called *completed likelihoods*, or *augmented likelihoods*. That is, we think of the likelihood over the data, $w_{1:N}$, as being completed by the realization of the latent variables $r_{1:N}$.

The completed likelihood is obtained by multiplying the generative model for the random variable and the measurement model

$$p(w_{1:N}, r_{1:N}|\theta) = p(w_{1:N}|r_{1:N}, \theta)p(r_{1:N}|\theta). \tag{3.17}$$

The true likelihood (sometimes called the incomplete data likelihood), by contrast, is $p(w_{1:N}|\theta)$. This likelihood is obtained from the completed likelihood as follows

$$p(w_{1:N}, r_{1:N}|\theta) = \int dr_1 \cdots dr_N p(w_{1:N}, r_{1:N}|\theta) \tag{3.18}$$

where the integral, over all allowed values of $r_{1:N}$, is substituted for a sum in dealing with discrete latent variables.

Example 3.3: Completed likelihoods for diffusive motion

In example 3.2, where we obtained an estimate for the diffusion coefficient, we assumed $w_n = r_n$. However, for a realistic Markov process involving measurement noise, the joint likelihood over $w_{1:N}$ and $r_{1:N}$ takes the form

$$p(w_{1:N}, r_{1:N}|D, \phi(r_{1:N})) = p(w_N|\phi(r_N))p(r_N|r_{N-1}, D) \tag{3.19}$$

$$\cdots p(w_2|\boldsymbol{\phi}_{r_2})p(r_2|r_1, D)p(w_1|\phi_{r_1})p(r_1) \tag{3.20}$$

$$= \left(\prod_{n=2}^{N} p(w_n|\phi(r_n))p(r_n|r_{n-1}, D)\right) p(w_1|r_1, \phi(r_1))p(r_1). \tag{3.21}$$

The object above is the completed likelihood. This likelihood cannot be interpreted as the probability of the observation. It is a joint distribution over the observed $w_{1:N}$ and unobserved (latent) states $r_{1:N}$.

The corresponding true likelihood is then obtained by explicitly marginalizing

$p(w_{1:N}, r_{1:N}|D, \phi(r_{1:N}))$ over the realizations $r_{1:N}$. In other words,

$$p(w_{1:N}|D, \phi) = \int dr_1 \cdots dr_N p(w_{1:N}, r_{1:N}|D, \phi(r_{1:N}))$$

where we have now dropped the subscripts over ϕ on the true likelihood shown on the left hand side.

Maximizing the likelihood $p(w_{1:N}|D, \phi)$ will yield an estimate for the diffusion, $\hat{D}$, which is different from ignoring the measurement noise altogether (*i.e.* assuming $w_n = r_n$). Indeed, we can already intuit that if our goal is to infer diffusion coefficients, D, those diffusion coefficients estimated by ignoring measurement noise will overestimate the true diffusion coefficient.

Put differently, if do not account for measurement noise, we overestimate the diffusion coefficient as both measurement noise and noise inherent to the diffusive process contribute positively to the apparent diffusion coefficient.

Above we discuss both completed and true likelihoods for a diffusion process. We also allude to how diffusion coefficient estimates may differ if we maximize the true likelihood by contrast to a likelihood ignoring a measurement model. Below we delve into this by returning to the specific example of a diffusing particle.

Example 3.4: True likelihood for normal diffusion

Returning to our simple example of a diffusing particle with Gaussian observations, note 3.2, we find

$$p(w_{1:N}|D, \sigma^2) = \int dr_1 \cdots dr_N p(w_{1:N}, r_{1:N}|D, \boldsymbol{\phi}_{r_{1:N}}) \tag{3.22}$$

$$= \frac{1}{(2\pi\sigma^2)^{N/2}} \cdot \frac{1}{(4\pi D\tau)^{(N-1)/2}} \tag{3.23}$$

$$\times \int dr_1 \cdots dr_N e^{-\frac{1}{2} r_{1:N} \cdot \boldsymbol{A} \cdot r_{1:N}^T + B w_{1:N} \cdot r_{1:N}^T - w_{1:N} \cdot \boldsymbol{C} \cdot w_{1:N}^T} p(r_1). \tag{3.24}$$

where $\boldsymbol{A}$ and $\boldsymbol{C}$ are matrices ($\boldsymbol{A}$ is tridiagonal and $\boldsymbol{C}$ is diagonal), B is a scalar and $r_{1:N}^T$ is the transpose of $r_{1:N}$. We can assume that $p(r_1)$ is a delta function or, for convenience, assume it is distributed as a Normal with a small variance, σ_1^2, whose mean is defined as the origin. In the latter case, the presence of $p(r_1)$ can be incorporated by updating the first column and row elements of $\boldsymbol{A}$. We call this new matrix $\boldsymbol{A}'$. Both $\boldsymbol{A}$ and $\boldsymbol{A}'$ depend on D and, thus, we write $\boldsymbol{A}' = \boldsymbol{A}'(D)$.

Assuming normally distributed initial conditions, we evaluate eq. (3.24) as follows

$$p(w_{1:N}|D, \sigma^2) = \frac{1}{(2\pi\sigma_1^2)^{1/2}} \cdot \frac{1}{(2\pi\sigma^2)^{N/2}} \cdot \frac{1}{(4\pi D\tau)^{(N-1)/2}} \tag{3.25}$$

$$\times \int dr_1 \cdots dr_N e^{-r_{1:N} \cdot \boldsymbol{A}'(D) \cdot r_{1:N}^T + B w_{1:N} \cdot r_{1:N}^T - w_{1:N} \cdot \boldsymbol{C} \cdot w_{1:N}^T} \tag{3.26}$$

$$\propto D^{-(N-1)/2} \cdot e^{-w_{1:N} \cdot \boldsymbol{C} \cdot w_{1:N}^T} \cdot \frac{1}{\sqrt{Det \boldsymbol{A}'(D)}} \cdot e^{B^2 w_{1:N} \cdot \boldsymbol{A}'^{-1}(D) \cdot w_{1:N}^T} \tag{3.27}$$

where only those portions of the prefactor depending on D have been retained.

We can now maximize this likelihood over D and the result is, expectedly, messy. To gain immediate intuition, it is worth simplifying to the case of just 2 data points

$$p(w_{1:2}|D, \sigma^2) = \int dr_1 dr_2 \frac{1}{(2\pi\sigma^2)} \cdot \frac{1}{(4\pi D\tau)^{1/2}} \tag{3.28}$$

$$\times e^{-\frac{(w_2 - r_2)^2}{2\sigma^2}} e^{-\frac{w_1^2}{2\sigma^2}} e^{-\frac{(r_2 - r_1)^2}{4D\tau}} p(r_1) \tag{3.29}$$

$$\propto \frac{1}{\sqrt{2D\tau + \sigma^2}} e^{-\frac{w_2^2}{2(2D\tau + \sigma^2)}}. \tag{3.30}$$

where all terms independent of D were dropped in the proportionality after integration and, for concreteness, $p(r_1)$ was assumed to be a delta function centered at the origin.

Taking the derivative of the log likelihood ($\log p(w_{1:2}|D,\sigma^2)$) with respect to D yields a linear equation. When solved for $\hat{D}$ it yields

$$\hat{D} = \frac{w_2^2}{2\tau} - \frac{\sigma^2}{\tau}. \tag{3.31}$$

Intuitively, this result makes sense. We find ourselves subtracting out a contribution from the diffusion coefficient estimate arising due to noise variance.

3.4.2 Likelihoods with missing observations

The concept of a likelihood can be pushed further to treat missing observations. For example, suppose observations $w_{i:j}$ from $w_{1:N}$ are missing. In other words, we have data $\{w_{1:i-1}, w_{j+1:N}\}$.

This problem is again treated very similarly to the problem above. Using a diffusing particle once more as an example, we have

$$p(\{w_{1:i-1}, w_{j+1:N}\}|D,\phi) = \int dw_i \cdots dw_j dr_1 \cdots dr_N \left(\prod_{n=2}^{N} p(w_n|\phi_{r_n}) p(r_i|r_{i-1}, D)\right) p(w_1|\phi_{r_1}) p(r_1). \tag{3.32}$$

Intuitively, the net effect of true likelihoods is to reduce the effective data set size from N to $N-(j-i)$ by assuming that the unknown measurements could have taken on any value. In effect, we already implicitly do this when considering likelihoods over $w_{1:N}$ rather than $w_{1:\infty}$ as we can think of likelihoods over observations $w_{1:N}$ as results of experiments for which the measurement index may have exceeded N but for which only the first N were ever recorded. As such, a likelihood over $w_{1:N}$ is equivalent to one where we have marginalized over realizations of experiments, $w_{N+1:\infty}$, never performed or recorded.

3.5 Likelihood maximization using the EM algorithm*

Likelihoods are key objects of frequentist inference. As we will see, they are also key objects of Bayesian inference that we will discuss in subsequent chapters.

As is often the case in any inference problem, we encounter measurement noise. Within the frequentist paradigm, this implies that our goal should be to maximize true likelihoods, $p(w_{1:N}|\theta)$, with respect to parameters θ (which include observation parameters ϕ).

In constructing completed likelihoods previously, we have had the advantage of being able to integrate over the latent variable $r_{1:N}$ and maximize the resulting marginal likelihood over θ. In general, both marginalizing over the latent variable as well maximizing over this likelihood analytically is challenging or even, most often, impossible. For this reason, here we focus on the problem of maximizing $p(w_{1:N}|\theta)$ under the assumption that marginalization over the latent variables in $p(w_{1:N}, r_{1:N}|\theta)$ is not analytically tractable.

The method we present to solve this problem is termed the *expectation-maximization algorithm* or EM for short. A brief motivation for this method is provided below.

Note 3.5: Motivation for EM

To show why it is difficult to maximize $p(w_{1:N}|\theta)$ directly, we start with

$$p(w_{1:N}|\theta) = \sum_{r_{1:N}} p(w_{1:N}, r_{1:N}|\theta) \tag{3.33}$$

where, for concreteness, we have assumed discrete latent variables, $r_{1:N}$, and the sum is over all allowed realizations of these latent variables.

*This is an advanced topic and could be skipped on a first reading.

We recall that, in order to maximize this likelihood and avoid numerical underflow, we need to maximize $\log\left(\sum_{r_{1:N}} p(w_{1:N}, r_{1:N}|\theta)\right)$. Maximizing the logarithm of the sum of a number of small terms (that may numerically underflow) is much more difficult than maximizing, say, the sum of the logarithm, $\sum_{r_{1:N}} \log p(w_{1:N}, r_{1:N}|\theta)$ where each term will not typically underflow. Unfortunately, these maxima are not in general equivalent.

The goal of the EM algorithm is ultimately to maximize an object expressible as the sum over logarithms as an approximation to maximizing over the logarithm of the sum.

The logic of EM is as follows: rather than to maximize $\log p(w_{1:N}|\theta)$, we can show that this is approximately equal to maximizing the expectation of $\log p(r_{1:N}, w_{1:N}|\theta)$ averaged over $p(r_{1:N}|w_{1:N}, \theta)$. The proof is provided below.

The objective is to maximize $p(w_{1:N}|\theta)$. This is equivalent to maximizing $\log p(w_{1:N}|\theta)$.

We start with

$$
\begin{aligned}
p\left(w_{1:N}, r_{1:N}|\theta\right) &= p\left(r_{1:N}|w_{1:N}, \theta\right) p\left(w_{1:N}|\theta\right) \\
\implies \log p\left(w_{1:N}, r_{1:N}|\theta\right) &= \log p\left(r_{1:N}|w_{1:N}, \theta\right) + \log p\left(w_{1:N}|\theta\right) \\
\implies \log p\left(w_{1:N}|\theta\right) &= \log p\left(w_{1:N}, r_{1:N}|\theta\right) - \log p\left(r_{1:N}|w_{1:N}, \theta\right) \\
\implies \left[\log p\left(w_{1:N}|\theta\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right) &= \left[\log p\left(w_{1:N}, r_{1:N}|\theta\right) - \log p\left(r_{1:N}|w_{1:N}, \theta\right)\right] \\
&\quad \times p\left(r_{1:N}|w_{1:N}, \theta^{old}\right) \\
\implies \sum_{r_{1:N}} \left[\log p\left(w_{1:N}|\theta\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right) &= \sum_{r_{1:N}} \left[\log p\left(w_{1:N}, r_{1:N}|\theta\right) - \log p\left(r_{1:N}|w_{1:N}, \theta\right)\right] \\
&\quad \times p\left(r_{1:N}|w_{1:N}, \theta^{old}\right) \\
\implies \log p\left(w_{1:N}|\theta\right) &= \underbrace{\sum_{r_{1:N}} \left[\log p\left(w_{1:N}, r_{1:N}|\theta\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right)}_{Q_{\theta^{old}}(\theta)} \\
&\quad - \sum_{r_{1:N}} \left[\log p\left(r_{1:N}|w_{1:N}, \theta\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right).
\end{aligned}
$$

So far, everything holds for all values of θ. In particular, for $\theta = \theta^{old}$ the last equality becomes

$$
\begin{aligned}
\log p\left(w_{1:N}|\theta^{old}\right) &= \underbrace{\sum_{r_{1:N}} \left[\log p\left(w_{1:N}, r_{1:N}|\theta^{old}\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right)}_{Q_{\theta^{old}}(\theta^{old})} \\
&\quad - \sum_{r_{1:N}} \left[\log p\left(r_{1:N}|w_{1:N}, \theta^{old}\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right).
\end{aligned} \tag{3.34}
$$

Subtracting $\log p\left(w_{1:N}|\theta^{old}\right)$ from $\log p\left(w_{1:N}|\theta\right)$, we have

$$
\begin{aligned}
&\log p\left(w_{1:N}|\theta\right) - \log p\left(w_{1:N}|\theta^{old}\right) \\
&= \underbrace{\sum_{r_{1:N}} \left[\log p\left(w_{1:N}, r_{1:N}|\theta\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right)}_{Q_{\theta^{old}}(\theta)} - \underbrace{\sum_{r_{1:N}} \left[\log p\left(w_{1:N}, r_{1:N}|\theta^{old}\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right)}_{Q_{\theta^{old}}(\theta^{old})} \\
&+ \sum_{r_{1:N}} \left[\log p\left(r_{1:N}|w_{1:N}, \theta^{old}\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right) - \sum_{r_{1:N}} \left[\log p\left(r_{1:N}|w_{1:N}, \theta\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right).
\end{aligned} \tag{3.35}
$$

Now, due to Gibbs' inequality, we know that

$$
\sum_{r_{1:N}} \left[\log p\left(r_{1:N}|w_{1:N}, \theta^{old}\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right) - \sum_{r_{1:N}} \left[\log p\left(r_{1:N}|w_{1:N}, \theta\right)\right] p\left(r_{1:N}|w_{1:N}, \theta^{old}\right) \leq 0 \tag{3.36}
$$

which implies

$$\begin{aligned}&\log p\left(w_{1:N}|\theta\right)-\log p\left(w_{1:N}|\theta^{old}\right)\\&\geq\underbrace{\sum_{r_{1:N}}\left[\log p\left(w_{1:N},r_{1:N}|\theta\right)\right]p\left(r_{1:N}|w_{1:N},\theta^{old}\right)}_{Q_{\theta^{old}}(\theta)}-\underbrace{\sum_{r_{1:N}}\left[\log p\left(w_{1:N},r_{1:N}|\theta^{old}\right)\right]p\left(r_{1:N}|w_{1:N},\theta^{old}\right)}_{Q_{\theta^{old}}(\theta^{old})}.\end{aligned}\tag{3.37}$$

Next, if a θ^{new} is selected such that $Q_{\theta^{old}}\left(\theta^{new}\right)\geq Q_{\theta^{old}}\left(\theta^{old}\right)$, we have $\log p\left(w_{1:N}|\theta^{new}\right)\geq\log p\left(w_{1:N}|\theta^{old}\right)$. This is guaranteed provided that

$$\theta^{new}=\underset{\theta}{\operatorname{argmax}}\,Q_{\theta^{old}}\left(\theta\right).\tag{3.38}$$

Computing $Q_{\theta^{old}}\left(\theta\right)$ is the expectation or "E-step" while maximizing $Q_{\theta^{old}}\left(\theta\right)$ is the maximization of "M-step". The E-step requires an expression for the sum or integral

$$Q_{\theta^{old}}\left(\theta\right)=\sum_{r_{1:N}}\left[\log p\left(w_{1:N},r_{1:N}|\theta\right)\right]p\left(r_{1:N}|w_{1:N},\theta^{old}\right)\tag{3.39}$$

which coincides with the expectation of $\log p\left(w_{1:N},r_{1:N}|\theta\right)$ with respect to $p\left(r_{1:N}|w_{1:N},\theta^{old}\right)$.

Algorithm 3.1: Expectation-Maximization

Given observations $w_{1:N}$, compute an approximation of the maximum-likelihood parameter values $\hat{\theta}=\operatorname{argmax}_\theta\, p(w_{1:N}|\theta)$ as follows

- Start at some θ^{old}
- Find the PDF of the conditional latent variables

$$f_{\theta^{\text{old}}}(r_{1:N})=p(r_{1:N}|w_{1:N},\theta^{\text{old}})$$

- Find the expectation of the complete log-likelihood

$$Q_{\theta^{\text{old}}}(\theta)=\sum_{r_{1:N}}f_{\theta^{\text{old}}}(r_{1:N})\log p(r_{1:N},w_{1:N}|\theta)$$

- Compute the new parameters values θ^{new} by the maximizer of $Q_{\theta^{\text{old}}}(\theta^{\text{new}})$
- Iterate until convergence.

In the long run, the following example should become a stand-alone "case study" section. I can image a biological setting that requires clustering. Because the components need to be at most 2 yet, it might not be hard to find an example in genetics where we need to clarify phenotypes involving 2 alleles.

Example 3.5: EM training of 2-Gaussian mixture

Consider scalar observations w_n, for $n=1,\ldots,N$, generated by a mixture of two Gaussians

$$w_n|\theta\sim\pi_1\text{Normal}(\mu_1,v_1)+\pi_2\text{Normal}(\mu_2,v_2)\tag{3.40}$$

where μ_1,μ_2 are the centers of the Normal and v_1,v_2 of both their variances, respectively. Here, π_1,π_2 are the probabilities of w_n stemming from the first and second Normal, respectively, so that they must satisfy $\pi_1+\pi_2=1$. Together, all parameters π_1,π_2,μ_1,μ_2 and v_1,v_2 are collected in θ.

In this example, we call our latent variables $s_{1:N}$. Here $s_{1:n}$ coincides with which of the two Gaussians generated the observation w_n and each s_n may take values 1 or 2 depending on weather w_n has been generated from Normal(μ_1,v_1) or Normal(μ_2,v_2), respectively.

With the introduction of these indicators, the model takes the equivalent form

$$s_n \sim \text{Categorical}_{1:2}\left(\pi_1, \pi_2\right) \tag{3.41}$$

$$w_n | s_n \sim \text{Normal}\left(\mu_{s_n}, v_{s_n}\right). \tag{3.42}$$

The $\text{Categorical}_{1,2}\left(\pi_1, \pi_2\right)$ distribution appearing in the first equation indicates that $p(s_n = 1) = \pi_1$ and $p(s_n = 2) = \pi_2$, which may be combined into a single expression

$$p(s_n|\theta) = \pi_1^{\delta_1(s_n)} \pi_2^{\delta_2(s_n)}. \tag{3.43}$$

To estimate the parameters $\theta = (\pi_1, \pi_2, \mu_1, \mu_2, v_1, v_2)$, as before, we seek a maximum likelihood solution which may be obtained by applying EM to the model of eqs. (3.41) and (3.42). In this model, the observations are $w_{1:N}$ and the latent variables are $s_{1:N}$.

The steps involved are as follows:

- We start from some initial guess of the parameters $\theta^{old} = (\pi_1^{old}, \pi_2^{old}, \mu_1^{old}, \mu_2^{old}, v_1^{old}, v_2^{old})$
- We then compute the conditional PDF, $f_{\theta^{old}}(s_{1:N})$, of the (iid) latent variables

$$f_{\theta^{old}}(s_{1:N}) = p\left(s_{1:N}|w_{1:N}, \theta^{old}\right) = \frac{p\left(s_{1:N}, w_{1:N}|\theta^{old}\right)}{p\left(w_{1:N}|\theta^{old}\right)} = \frac{p\left(w_{1:N}|s_{1:N}, \theta^{old}\right)}{p\left(w_{1:N}|\theta^{old}\right)} p\left(s_{1:N}|\theta^{old}\right) \tag{3.44}$$

$$= \prod_{n=1}^{N} \left[\frac{p\left(w_n|s_n, \theta^{old}\right)}{p\left(w_n|\theta^{old}\right)} p\left(s_n|\theta^{old}\right) \right] \tag{3.45}$$

$$= \prod_{n=1}^{N} \left[\frac{\text{Normal}\left(w_n; \mu_{s_n}^{old}, v_{s_n}^{old}\right) \left(\pi_1^{old}\right)^{\delta_1(s_n)} \left(\pi_2^{old}\right)^{\delta_2(s_n)}}{\pi_1^{old}\text{Normal}\left(w_n; \mu_1^{old}, v_1^{old}\right) + \pi_2^{old}\text{Normal}\left(w_n; \mu_2^{old}, v_2^{old}\right)} \right] \tag{3.46}$$

$$= \prod_{n=1}^{N} \left[\frac{\left(\pi_1^{old}\text{Normal}\left(w_n; \mu_1^{old}, v_1^{old}\right)\right)^{\delta_1(s_n)} \left(\pi_2^{old}\text{Normal}\left(w_n; \mu_2^{old}, v_2^{old}\right)\right)^{\delta_2(s_n)}}{\pi_1^{old}\text{Normal}\left(w_n; \mu_1^{old}, v_1^{old}\right) + \pi_2^{old}\text{Normal}\left(w_n; \mu_2^{old}, v_2^{old}\right)} \right] \tag{3.47}$$

$$= \prod_{n=1}^{N} \left[\left(\frac{\pi_1^{old}\text{Normal}\left(w_n; \mu_1^{old}, v_1^{old}\right)}{\pi_1^{old}\text{Normal}\left(w_n; \mu_1^{old}, v_1^{old}\right) + \pi_2^{old}\text{Normal}\left(w_n; \mu_2^{old}, v_2^{old}\right)} \right)^{\delta_1(s_n)} \right. \tag{3.48}$$

$$\left. \times \left(\frac{\pi_2^{old}\text{Normal}\left(w_n; \mu_2^{old}, v_2^{old}\right)}{\pi_1^{old}\text{Normal}\left(w_n; \mu_1^{old}, v_1^{old}\right) + \pi_2^{old}\text{Normal}\left(w_n; \mu_2^{old}, v_2^{old}\right)} \right)^{\delta_2(s_n)} \right] \tag{3.49}$$

$$= \prod_{n=1}^{N} \left(\gamma_{1n}^{old}\right)^{\delta_1(s_n)} \left(\gamma_{2n}^{old}\right)^{\delta_2(s_n)} \tag{3.50}$$

where

$$\gamma_{1n}^{old} = \frac{\pi_1^{old}\text{Normal}\left(w_n; \mu_1^{old}, v_1^{old}\right)}{\pi_1^{old}\text{Normal}\left(w_n; \mu_1^{old}, v_1^{old}\right) + \pi_2^{old}\text{Normal}\left(w_n; \mu_2^{old}, v_2^{old}\right)} \tag{3.51}$$

$$\gamma_{2n}^{old} = \frac{\pi_2^{old}\text{Normal}\left(w_n; \mu_2^{old}, v_2^{old}\right)}{\pi_1^{old}\text{Normal}\left(w_n; \mu_1^{old}, v_1^{old}\right) + \pi_2^{old}\text{Normal}\left(w_n; \mu_2^{old}, v_2^{old}\right)}. \tag{3.52}$$

These are essentially the probabilities of observation w_n stemming from the first and second Gaussians in the mixture, respectively, under the parameter values θ^{old}.

• Next, we compute the expectation under $f_{\theta^{old}}(s_{1:N})$ of the complete log-likelihood

$$Q_{\theta^{old}}(\theta) = \sum_{s_{1:N}} \log p(s_{1:N}, w_{1:N}|\theta) f_{\theta^{old}}(s_{1:N}) \tag{3.53}$$

$$= \sum_{s_{1:N}} \left[\log p(s_{1:N}, w_{1:N}|\theta) \prod_{n=1}^{N} (\gamma_{1n}^{old})^{\delta_1(s_n)} (\gamma_{2n}^{old})^{\delta_2(s_n)} \right] \tag{3.54}$$

$$= \sum_{s_{1:N}} \left[\sum_{n'=1}^{N} [\log p(s_{n'}, w_{n'}|\theta)] \prod_{n=1}^{N} \left[(\gamma_{1n}^{old})^{\delta_1(s_n)} (\gamma_{2n}^{old})^{\delta_2(s_n)} \right] \right] \tag{3.55}$$

$$= \sum_{n=1}^{N} \left[\sum_{s_n=1}^{2} (\gamma_{1n}^{old})^{\delta_1(s_n)} (\gamma_{2n}^{old})^{\delta_2(s_n)} \log p(s_n, w_n|\theta) \right] \tag{3.56}$$

$$= \sum_{n=1}^{N} \left[\sum_{s_n=1}^{2} (\gamma_{1n}^{old})^{\delta_1(s_n)} (\gamma_{2n}^{old})^{\delta_2(s_n)} [\log p(w_n|s_n, \theta) p(s_n|\theta)] \right] \tag{3.57}$$

$$= \sum_{n=1}^{N} \left[\sum_{s_n=1}^{2} (\gamma_{1n}^{old} [\log \mathsf{Normal}(w_n; \mu_1, v_1)\pi_1])^{\delta_1(s_n)} (\gamma_{2n}^{old} [\log \mathsf{Normal}(w_n; \mu_2, v_2)\pi_2])^{\delta_2(s_n)} \right] \tag{3.58}$$

$$= \sum_{n=1}^{N} \left[\gamma_{1n}^{old} [\log \mathsf{Normal}(w_n; \mu_1, v_1)\pi_1] + \gamma_{2n}^{old} [\log \mathsf{Normal}(w_n; \mu_2, v_2)\pi_2] \right] \tag{3.59}$$

$$= \sum_{n=1}^{N} \left[\gamma_{1n}^{old} \left(\log \pi_1 - \frac{\log(2\pi v_1)}{2} - \frac{(w_n - \mu_1)^2}{2v_1} \right) \right] \tag{3.60}$$

$$+ \sum_{n=1}^{N} \left[\gamma_{2n}^{old} \left(\log \pi_2 - \frac{\log(2\pi v_2)}{2} - \frac{(w_n - \mu_2)^2}{2v_2}] \right) \right] \tag{3.61}$$

$$= \sum_{n=1}^{N} \left[\gamma_{1n}^{old} \left(\log \pi_1 - \frac{\log v_1}{2} - \frac{(w_n - \mu_1)^2}{2v_1} \right) \right] \tag{3.62}$$

$$+ \sum_{n=1}^{N} \left[\gamma_{2n}^{old} \left(\log \pi_2 - \frac{\log v_2}{2} - \frac{(w_n - \mu_2)^2}{2v_2} \right) \right] + \text{constants} \tag{3.63}$$

$$= \left(\log \pi_1 - \frac{\log v_1}{2} \right) \left(\sum_{n=1}^{N} \gamma_{1n}^{old} \right) + \left(\log \pi_2 - \frac{\log v_2}{2} \right) \left(\sum_{n=1}^{N} \gamma_{2n}^{old} \right) \tag{3.64}$$

$$- \frac{1}{2} \sum_{n=1}^{N} \left[\gamma_{1n}^{old} \frac{(w_n - \mu_1)^2}{v_1} + \gamma_{2n}^{old} \frac{(w_n - \mu_2)^2}{v_2} \right] + \text{constants} \tag{3.65}$$

• Finally, the objective is to find the maximizer $\theta^{new} = (\pi_1^{new}, \pi_2^{new}, \mu_1^{new}, \mu_2^{new}, v_1^{new}, v_2^{new})$ of $Q_{\theta^{old}}(\theta)$ under the constraint

$$\pi_1 + \pi_2 = 1. \tag{3.66}$$

For this, consider a Lagrangian

$$L_{\theta^{old}}(\theta, \lambda) = Q_{\theta^{old}}(\theta) + \lambda(\pi_1 + \pi_2 - 1) \tag{3.67}$$

with multiplier λ, that is used to enforce the constraint on the weights. Equating the gradient of $L_{\theta^{old}}(\theta)$ to zero,

we obtain

$$0 = \frac{\partial L_{\theta^{old}}(\theta, \lambda)}{\partial \pi_1} = \frac{1}{\pi_1}\left(\sum_{n=1}^{N} \gamma_{1n}^{old}\right) + \lambda \tag{3.68}$$

$$0 = \frac{\partial L_{\theta^{old}}(\theta, \lambda)}{\partial \pi_2} = \frac{1}{\pi_2}\left(\sum_{n=1}^{N} \gamma_{2n}^{old}\right) + \lambda \tag{3.69}$$

which, combined with the constraint $\pi_1 + \pi_2 = 1$, yield the optimum

$$\pi_1^{new} = \frac{\sum_{n=1}^{N} \gamma_{1n}^{old}}{\sum_{n=1}^{N} (\gamma_{1n}^{old} + \gamma_{2n}^{old})} \tag{3.70}$$

$$\pi_2^{new} = \frac{\sum_{n=1}^{N} \gamma_{2n}^{old}}{\sum_{n=1}^{N} (\gamma_{1n}^{old} + \gamma_{2n}^{old})} \tag{3.71}$$

From the remaining components of the gradient, we obtain

$$0 = \frac{\partial L_{\theta^{old}}(\theta, \lambda)}{\partial \mu_1} = \sum_{n=1}^{N} \gamma_{1n}^{old} \frac{(w_n - \mu_1)}{v_1} \tag{3.72}$$

$$0 = \frac{\partial L_{\theta^{old}}(\theta, \lambda)}{\partial \mu_2} = \sum_{n=1}^{N} \gamma_{1n}^{old} \frac{(w_n - \mu_2)}{v_2} \tag{3.73}$$

$$0 = \frac{\partial L_{\theta^{old}}(\theta, \lambda)}{\partial v_1} = \frac{1}{2v_1}\left(\frac{1}{v_1}\left(\sum_{n=1}^{N} \gamma_{1n}^{old}(w_n - \mu_1)^2\right) - \left(\sum_{n=1}^{N} \gamma_{1n}^{old}\right)\right) \tag{3.74}$$

$$0 = \frac{\partial L_{\theta^{old}}(\theta, \lambda)}{\partial v_2} = \frac{1}{2v_2}\left(\frac{1}{v_2}\left(\sum_{n=1}^{N} \gamma_{2n}^{old}(w_n - \mu_2)^2\right) - \left(\sum_{n=1}^{N} \gamma_{2n}^{old}\right)\right) \tag{3.75}$$

which can be solved to yield the remaining parameters

$$\mu_1^{new} = \frac{\sum_{n=1}^{N} \gamma_{1n}^{old} w_n}{\sum_{n=1}^{N} \gamma_{1n}^{old}} \tag{3.76}$$

$$\mu_2^{new} = \frac{\sum_{n=1}^{N} \gamma_{2n}^{old} w_n}{\sum_{n=1}^{N} \gamma_{2n}^{old}} \tag{3.77}$$

$$v_1^{new} = \frac{\sum_{n=1}^{N} \gamma_{1n}^{old} (w_n - \mu_1^{new})^2}{\sum_{n=1}^{N} \gamma_{1n}^{old}} \tag{3.78}$$

$$v_2^{new} = \frac{\sum_{n=1}^{N} \gamma_{2n}^{old} (w_n - \mu_2^{new})^2}{\sum_{n=1}^{N} \gamma_{2n}^{old}}. \tag{3.79}$$

3.6 Exercise problems

Exercise 3.1: Birth-death process likelihoods in continuous time

Consider initially a birth process alone.

1. Simulate a realization of the birth process using the Gillespie algorithm.
2. Use the results of the simulation to write down the likelihood of this sequence of realizations as they are observed in continuous time. Hint: The likelihood will be the product of terms that coincide with the probability that no event occur over the time between events and that an event occur over the infinitesimal

interval around the time at which an event occurs.
3. Maximize your likelihood to obtain the birth rate. How does the breadth of your log likelihood change as your data set grows? As we will see in a later example, the breadth of the log likelihood is related to a measure of uncertainty in frequentist estimate of the birth rate.
4. Repeat the steps above for the birth-death process (with a death rate a tenth of the birth rate) in order to obtain both birth and death rates.

Exercise 3.2: Birth-death process likelihoods in discrete time

Here we reconsider the preceding example but assume that measurements are provided at regular and discrete time intervals. These regular time intervals can be shorter or, more interestingly, either on par or longer than the typical time it takes for birth and death events to occur.

1. Simulate a realization of the birth-death process using the Gillespie algorithm with a death rate a tenth of the birth rate. Then create an array, $w_{1:N}$, coinciding with the population at discrete time levels. Initially, choose these time levels to be on par with the inverse birth rate.
2. Use the results of the simulation to write down the likelihood of this sequence of realizations as they are observed in discrete time. Hint: You will need to solve the master equation for this portion of the problem.
3. Maximize your likelihood to obtain both birth and death rates.
4. What happens to your birth and death rate estimates as the time levels become twice, three times and ten times as long as the inverse birth rate?

Exercise 3.3: Ornstein-Uhlenbeck process

Optical trapping, a technology earning the 2018 Physic Nobel Prize, can be used to confine micron-sized colloids, that would otherwise freely diffuse in solution, in an approximately harmonic potential. Here we model the optical trap in one-dimension.

1. Simulate an Ornstein-Uhlenbeck process and store the positions at regular time levels in an array $w_{1:N}$. That is, simulate the position of a particle confined by a harmonic trap but otherwise jostled by thermal fluctuations.
2. Use the results of the simulation to write down the likelihood of the sequence of realizations. Hint: you will need to solve the coinciding Fokker-Planck equation.
3. Maximize your likelihood to obtain the trap potential's harmonic force constant and the particle's friction coefficient.
4. Repeat the procedure assuming a Gaussian observation noise model. That is, take your original $w_{1:N}$ and add white noise to each of them to obtain the new sequence $w'_{1:N}$. Use this new sequence to learn the harmonic force constant, friction coefficient and the variance of the observation distribution.
5. How do your estimates for the friction coefficient and the harmonic force constant change if you re-analyze $w'_{1:N}$ under the assumption that no measurement noise is present?

Exercise 3.4: EM algorithm

Implement the EM algorithm for the simple case of a mixture of two Gaussians in 1D with identical standard deviation. In other words, begin by generating synthetic data according to this model: $\pi_1 e^{-\frac{(x-\mu_1)^2}{2\sigma^2}} + (1-\pi_1)e^{-\frac{(x-\mu_2)^2}{2\sigma^2}}$ where you have pre-specified by hand the parameters $\pi_1, \mu_1, \mu_2, \sigma$.
Then, implement an EM in order to learn the parameters $\pi_1, \mu_1, \mu_2, \sigma$. Compare your parameter estimates from EM to the theoretical values you used to generate your data.

Exercise 3.5: Cramer-Rao Lower Bound

An error bar around a maximum likelihood estimate can sometimes be approximated using a Cramer-Rao lower bound (CRLB). Briefly, under some approximations, the lower bound on the variance around a frequentist estimate, $\hat{\theta}$, is given by the inverse of the expectation of $-\partial^2 \log p(w_{1:N}|\theta)/\partial\theta^2$ with respect to $p(w_{1:N}|\theta)$ for a one-

dimensional parameter θ. That is,

$$var(\hat{\theta}) \geq -\left(\langle \partial^2 \log p(w_{1:N}|\theta)/\partial\theta^2 \rangle\right)^{-1}.$$

where $\langle\cdot\rangle$ denote the expectation with respect to the appropriate distribution.

1. Compute the CRLB for N Gaussian iid realizations. On this basis, provide an intuitive interpretation for $-\left(\langle \partial^2 \log p(w_{1:N}|\theta)/\partial\theta^2 \rangle\right)^{-1}$ assuming a unimodal likelihood.
2. Compute the CRLB for N Beta distribution iid realizations with $\alpha = \beta = 1/2$. Note that this distribution has two maxima. Is the CRLB useful in estimating the uncertainty around the frequentist estimate? Why or why not?

Chapter 4

Bayesian inference

By the end of this chapter, we will have presented
- *The concept of priors and posteriors*
- *Model selection with examples*
- *Graphical representations*

4.1 Modeling in Bayesian terms

In the previous chapters, we discussed common models of physical systems including observation models in the form of emission distributions. We then described how to construct likelihoods and obtain parameter estimates through likelihood maximization. This approach, often termed frequentist, relies on the notion of probabilities as frequencies of events. As such, it finds parameter values for models most consistent with experimental outcomes.

Within this frequentist approach, data are represented as random variables and model parameters are represented as numbers to be determined through likelihood maximization. In reality, likelihoods provide more than just parameter *point estimates* (parameter values coinciding with the likelihood's maximum). The curvature of the likelihood around its maximum tells us something about a parameter's uncertainty bound as well and this concept is briefly explored in a problem of the last chapter where we discuss the Cramer-Rao lower bound. However, while uncertainty bounds are indeed helpful, these fall short of telling us how a parameter is distributed. To determine how parameters are distributed, we first need to abandon the concept of parameters as numbers and treat them as random variables.

Thinking of parameters as random variables is also helpful in the case that the likelihood itself has multiple maxima. For example, if the likelihood has degenerate maxima over a parameter, maximum likelihood cannot tell us which of these is the appropriate point estimate. On the other hand, thinking of the parameters as random variables instead suggests that parameters themselves may be distributed and, as such, many possible parameter values may be warranted by the data.

4.1.1 The posterior distribution

Bayesian methods, the focus of this chapter, treat both data and model parameters as random variables. Of central importance in Bayesian analysis is the *posterior*, $p(\theta|w_{1:N})$. The posterior is the distribution over the model parameters θ informed by the data $w_{1:N}$.

Intuitively, we think of the data as refining our knowledge of the unknown parameters distributed according to their posterior. In other words, we imagine that given more and more data, our knowledge of θ improves and the posterior sharpens around those values of the parameters that generated the data.

These notions suggest that knowledge of parameters can be used to update this parameter's posterior as more data becomes available. That is, a posterior informed by $N-1$ data points, $p(\theta|w_{1:N-1})$, can be used to obtain an updated posterior $p(\theta|w_{1:N})$ as the N^{th} data point becomes available. A likelihood, dictated by a

generative model including measurement models, therefore helps us link subsequent posteriors as we gather more data. Logically we can write

$$p(\theta|w_{1:N}) \Longleftarrow p(\theta|w_{1:N-1}) \Longleftarrow \cdots \Longleftarrow p(\theta|w_1) \Longleftarrow ?$$

The question mark arises as, in the absence of data, we can ask how updating can even begin? Put differently, what is the $p(\theta|\cdot)$ used in the update below?

$$p(\theta|w_1) \propto p(w_1|\theta)p(\theta|\cdot).$$

Normally, we would have data on the right hand side of the conditional in $p(\theta|\cdot)$ and we will detail what could stand in its place shortly.

This probability density, reflecting our prior belief as to how the θ are distributed, is called the *prior*. That is, the distribution obtained prior to the data by analogy to the posterior obtained after the data.

Thus, the posterior distribution is proportional to the product of the prior and the likelihood and follows from Bayes' theorem

$$p(\theta|w_{1:N}) = \frac{p(w_{1:N}|\theta)p(\theta|\cdot)}{p(w_{1:N})} \tag{4.1}$$

where $p(w_{1:N})$ is obtained by completion

$$p(w_{1:N}) = \int d\theta\, p(\theta, w_{1:N}) = \int d\theta\, p(w_{1:N}|\theta)p(\theta|\cdot)$$

or normalization $\int d\theta\, p(\theta|w_{1:N}) = 1$.

The posterior $p(\theta|w_{1:N})$ exists as long as the denominator $p(w_{1:N})$, sometimes called the *evidence*, only insofar as the data can be generated from the model, *i.e.* that $p(w_{1:N}|\theta) \neq 0$. As usual, the integral above is understood as a sum for discrete parameters.

Note 4.1: Use of proportionality in Bayesian methods

We make a note on the use of the proportionality constant here as it is widely used in Bayesian calculations. We recall that the posterior is written as $p(\theta|w_{1:N})$. This implies that the posterior is normalized over all values of θ.

Thus, the proportionality constant, *i.e.* the normalization constant, is independent of θ and we can write

$$p(\theta|w_{1:N}) = \frac{p(w_{1:N}|\theta)p(\theta|\cdot)}{p(w_{1:N})}$$
$$p(\theta|w_{1:N}) \propto p(w_{1:N}|\theta)p(\theta|\cdot)$$

where $p(w_{1:N})^{-1}$ here is a normalization constant.

To be clear, the proportionality constant is not constant with respect to the data. Rather, it is constant with respect to parameters.

As a simple example of how to use priors and likelihoods, we use simple Bernoulli trials, *i.e.* coin flips, below as an illustration.

Example 4.1: Bernoulli trials within the Bayesian paradigm

Here we return to a simple example involving Bernoulli trials such as a coin flip. We label π the probability of the first of two outcomes, say heads. The likelihood of having collected N_H heads and N_T tails with $N_H + N_T = N$ trials, is

$$p(w_{1:N}|\pi) \propto \pi^{N_H}(1-\pi)^{N_T}.$$

For now, we assume a prior of the form

$$p(\pi|\alpha,\beta) \propto \pi^{\alpha-1}(1-\pi)^{\beta-1} \tag{4.2}$$

where α and β are positive constants. According to eq. (4.1), the product of the likelihood and prior yields, up to a proportionality constant independent of π, the posterior

$$p(\pi|w_{1:N}) \propto \pi^{N_H+\alpha-1}(1-\pi)^{N_T+\beta-1}.$$

We immediately note the difference between the maximum of the likelihood, obtained by maximizing the likelihood over π, namely $\text{argmax}_\pi p(w_{1:N}|\pi)$ which yields $\hat{\pi} = \frac{N_H}{N}$.

While Bayesian methods return the full posterior, we can also choose to maximize the posterior. The estimate for π obtained in this case is termed the *maximum a posteriori estimate* and it follows from $\text{argmax}_\pi p(\pi|w_{1:N})$. This operation yields $\hat{\pi} = (N_H + \alpha - 1)/(N + \alpha + \beta - 2)$.

Intuitively, here, we see that, on account of the special mathematical form we selected for the prior, the prior added *pseudocounts* to our measurement. That is, the prior added an additional $\alpha - 1$ counts to heads and $\beta - 1$ counts to tails. This interpretation is a welcome consequence of the special choice of prior that we made.

These results warrant a few remarks. First, even for this special choice of Bernoulli trials and Beta priors, we see that as the number of data points available grow, irrespective of our choice of α and β, our posterior eventually peaks for the same value of π as the likelihood.

Next, we also see a shortcoming of maximum likelihood, *i.e.* assuming that the probability of heads is 0 if no heads are drawn over the course of an experiment, are remedied by the prior. Indeed even after just one draw, in favor of tails say, the maximum a posteriori probability of heads is not 0. In fact, it is

$$\hat{\pi} = \frac{\alpha}{\alpha + \beta - 1}.$$

The distribution eq. (4.2) is the Beta distribution and it will re-appear often across chapters. The normalization constant omitted in eq. (4.2) is obtained by integrating $\pi^{\alpha-1}(1-\pi)^{\beta-1}$ with respect to π from 0 to one. The resulting integral, not shown here, is the Beta function.

The example above motivates a remark we now make on the importance of likelihoods. Priors essentially define the range over which parameters can be assigned non-zero posterior probability after data become available. It is reasonable to expect that, in the limit of a large N, our choice of prior becomes increasingly immaterial and the shape of the posterior is ultimately dictated by the likelihood.

This logic is illustrated in fig. 4.1. Here, for sufficiently independent observations, the likelihood's breadth eventually narrows with respect to its mean. The same is not true of the prior that is independent of the amount of data. Thus, eventually, with enough data, the likelihood dominates over the prior. This really drives home two points: 1) we must have likelihoods that capture both details of the physical system and the observation model; 2) there is danger in attempting parameter estimation with insufficient or poor data quality. In the latter case, our arbitrary choice of prior may deeply influence the ultimate shape of the posterior.

This discussion begs the question: how much data is enough to overcome the effects of the prior? Of course, it depends. It depends on the mathematical form for the likelihood, it depends on the quality of the data, it also depends on the "strength" of the prior. That is, how firm our prior beliefs are and how easily the likelihood can help shift the posterior away from the form of the prior.

Equivalently, how much data is enough to ascertain the optimality of one parameter choice over another to explain the data? The latter question brings us to the topic of the Bayes' factor discussed below.

Note 4.2: Bayes' factor

There is more to Bayes' theorem than computing or sampling from full posteriors. Often, we may only be interested in comparing two hypotheses. That is, we may be interested in evaluating the posterior at two different model

parameters values, say θ^1 and θ^2, given data $w_{1:N}$. The ratio of these posteriors is written as

$$\frac{p(\theta = \theta^1|w_{1:N})}{p(\theta = \theta^2|w_{1:N})}. \tag{4.3}$$

More generally, the posterior probability over disparate parameter ranges can also be compared by integrating the posterior over different parameter ranges and taking the ratio of the ensuing probabilities.

For uniform priors, the posterior ratio reduces to a *likelihood ratio* also called a *Bayes' factor*

$$\frac{p(w_{1:N}|\theta = \theta^1)}{p(w_{1:N}|\theta = \theta^2)} \tag{4.4}$$

often used to compare two parameter values.

One advantage of the posterior or likelihood ratio is that it typically becomes very large (or very small) as more data are amassed. This is because likelihoods, which for iid samples are constructed from products of likelihoods over single data point, are sensitive to small parameter variations. Indeed, likelihoods can quickly become vanishingly small for parameter values only slightly less optimal than others as the number of data points increases. For this reason, it is commonplace to compute logarithms of posterior or likelihood ratios. The sensitivity of likelihoods can also be a disadvantage: data point outliers can sharply reduce the likelihood.

4.1.2 The predictive distribution

The predictive distribution must be introduced here.

The predictive distribution of a Bayesian model is $p(w_{N+1}|w_{1:N})$. This distribution assumes that data $w_{1:N}$ are known and seeks to quantify the statistics of the next measurement w_{N+1}.

Basically this section needs to contain a motivating example *Why the predictive distribution is important?* and then pretty much the same discussion from section 4.5. The idea is to introduce notions here and then in section 4.5 focus only on methods and shortcomings.

4.1.3 Bayesian data analysis

Here provide the big abstract picture.

Note 4.3: The Bayesian paradigm of Data Analysis

In a Bayesian problem, we follow the usual pattern:

- We write down the likelihood of the data.
- We define a domain spanned by the model parameters and assign prior probability over this domain. We call this probability a *prior* distribution.
- We use Bayes' rule to update the prior using the likelihood to construct the posterior distribution over all parameters.

Conclude with a superficial discussion of pros and cons.

A note *The economy of data* would be appropriate here too.

4.2 Priors

We begin with a discussion of priors. Not because they are the most important (likelihoods contain all the relevant physics), but because they are the newest addition to our discussion.

There are two types of priors: uninformative and informative priors. Uninformative priors intuitively meet our expectation for what priors should be like and, for this reason, we start with these.

4.2.1 Uninformative priors

The simplest *uninformative prior* is inspired from Laplace's principle of insufficient reason when the set of hypotheses are complete and mutually exclusive. That is $p(\theta) =$ constant where the constant is independent of θ. This distribution is termed flat or uniform for a distribution over a continuous parameter over some allowed parameter range. When speaking of discrete distributions, we speak of equiprobable *a priori* outcomes.

Under the assumption that $p(\theta)$ is constant over some parameter range, the posterior and likelihood are directly related

$$p(\theta|w_{1:N}) \propto p(w_{1:N}|\theta)p(\theta) \propto p(w_{1:N}|\theta).$$

The constants of proportionality dropped are independent of θ. Therefore, the dependence of the likelihood and the posterior on θ is identical and, consequently, maximizing the posterior or maximizing the likelihood over θ results in identical parameter estimates, $\hat{\theta}$.

As intuitive as it may appear to start with flat priors, these can exhibit pathologies.

First, flat priors can really only be constructed for *bounded* continuous parameters. For example, a flat prior over the entire real line is improper, *i.e.* it does not normalize to 1. In fact, for a parameter whose range is anywhere from $[0, \infty]$, the flat prior cannot be normalized at all; $\int_0^\infty d\theta p(\theta) = \infty$. Thus $p(\theta)$ is not a probability distribution at all. Such priors are termed *improper priors* and have serious implications for more advanced Bayesian applications.

Second, a flat prior over a model parameter, say θ, over the interval [0, 1] is not quite as uninformative as it may appear as a coordinate transformation to an alternative variable, say e^θ, reveals that we suddenly know more about the random variable e^θ than we did about θ since its distribution over e^θ is no longer flat. Conversely, if e^θ is uniform on the interval [0, 1], then θ_1 becomes more concentrated at the upper boundary, 1.

A desire to make priors invariant under continuous variable transformation motivated the development of the *Jeffreys prior*. Other concerns, in turn, motivated other uninformative priors such as those used in statistical mechanics, *e.g.* the multinomial distribution and the related Shannon entropy. We do not dwell on these here.

Fundamentally such uninformative priors were conceived in an effort to enforce fundamental system properties or symmetries *a priori* irrespective of the associated computational cost of enforcing these. With large amounts of data often used in parameter estimation, it is no longer clear that the vanishingly small effect the prior has on the final posterior ultimately warrants the high computational cost introduced by these priors.

Add an example here with a discussion of a flat prior for the Bernoulli trial. This is Beta$(1, 1)$

In the long run we need a note with improper priors here

4.2.2 Informative priors

It may appear counter-intuitive to start with a prior that has some structure. That is, one that is not completely flat over the domain of interest. Yet, as we will see shortly, informative priors can be quite powerful.

A common choice of *informative prior* is directly suggested by the form of the likelihood. To see this, we reconsider how additionally available data is used to update a posterior. A new posterior, $p(\theta|w_2, w_1)$, is obtained from the old posterior, $p(\theta|w_1)$, and the likelihood as follows

$$p(\theta|w_2, w_1) \propto p(w_2|\theta, w_1)p(\theta|w_1).$$

In this way, the old posterior, $p(\theta|w_1)$, plays the role of the prior for the new posterior, $p(\theta|w_2, w_1)$. Intuitively, as more and more data become available, we expect the shape of the posterior to stop changing and we expect this shape to be dictated by the form of the likelihood.

4.2.3 Maximum entropy priors*

4.3 The logistics of Bayesian formulations

Let's be specific. Conjugate priors does not exist. Only conjugate pairs of prior-likelihood. It is better to talk about "conjugate models" or "conjugate formulations" or something similar.

*This is an advanced topic and could be skipped on a first reading.

4.3.1 Hierarchical Bayesian formulation

Introduce models with more than one layer of latent variables here

Note 4.4: Hyper-parameters

This note needs to move Whether conjugate or not, priors are distributions. Just like any other distribution, they have parameters and these are called *hyperparameters* . These hyperparameters are what appear on the right hand side of the conditional in the prior $p(\theta|\cdot)$.

Suppose we have hyperparameters γ, then we can write $p(\theta|\gamma)$. These hyperparameters, in turn, can also be distributed, *e.g.* according to the *hyperprior* $p(\gamma|\eta)$ where η are called *hyperhyperparameters*, thereby establishing a hierarchy of random variable dependencies.

Putting it all together, starting left to right, we would say that observations probabilistically depend on latent variables, latent variables probabilistically depend on model parameters, model parameters probabilistically depend on hyperparameters, and so forth.

4.3.2 Conjugate Bayesian formulation

Put differently, if we insist that the prior and all future posteriors retain the same mathematical form, then the mathematical form for the likelihood fixes the shape of the prior. We call these priors *c*onjugate priors and conjugate priors attain analytical forms when the likelihood belongs to a very general family of distributions belonging to the *e*xponential class that we discuss in greater depth shortly.

It turns out that the concept of conjugacy was already introduced in the example provided on Bernoulli trials where the likelihood is a Bernoulli distribution. The conjugate to the Bernoulli is the Beta distribution. Below, we discuss another example of conjugacy before turning to a formal description.

Example 4.2: Determining the prior conjugate to a Poisson likelihood

To illustrate the concept of conjugacy, we consider an experiment whose likelihood for the n^{th} event is well described by a Poisson distribution

$$p(w_n|\lambda) = \frac{\lambda^{w_n}}{w_n!}e^{-\lambda}. \tag{4.5}$$

Here λ is understood as a unitless parameter, such as a rate multiplied by a time.

The conjugate prior, $p(\lambda|\cdot)$, to the Poisson likelihood is the Gamma distribution

$$p(\lambda|\alpha,\beta) = \text{Gamma}(\lambda;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda} \tag{4.6}$$

which contains two hyperparameters, α and β.

To verify that the Gamma distribution is indeed conjugate to the Poisson, we consider the posterior after just one observation of w_1. The posterior is the product of the likelihood and the prior and reads

$$p(\lambda|w_1,\alpha,\beta) \propto \lambda^{w_1}e^{-\lambda} \times \lambda^{\alpha-1}e^{-\beta\lambda} \propto \text{Gamma}(\lambda; w_1+\alpha, 1+\beta). \tag{4.7}$$

In other words, the posterior is also a Gamma distribution, just like the prior. After N independent measurements, with $w_{1:N} = \{w_1, \cdots w_N\}$, we have

$$p(\lambda|w_{1:N},\alpha,\beta) = \text{Gamma}\left(\lambda; \sum_{n=1}^{N} w_n + \alpha, N+\beta\right). \tag{4.8}$$

Fig. (4.1) illustrates how the posterior is dominated by the likelihood provided sufficient data and how an arbitrary choice of hyperparameters becomes less important for large enough N.

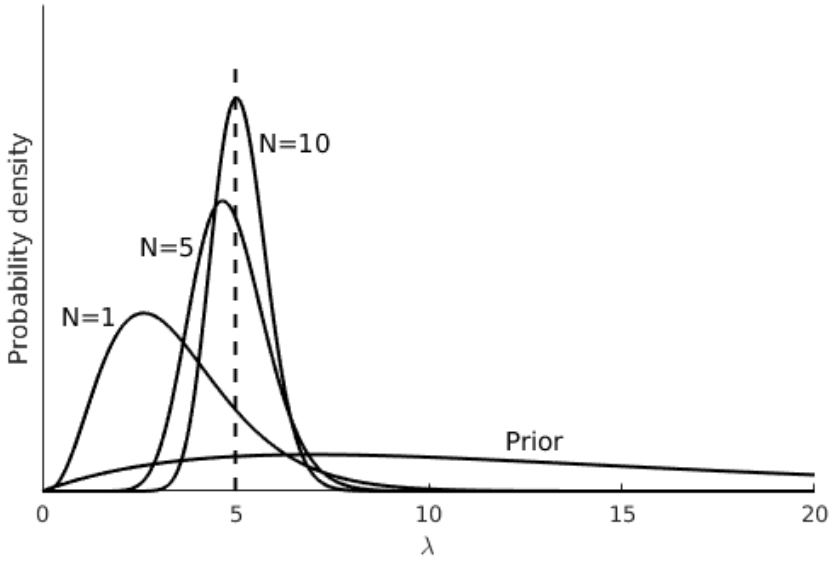


Figure 4.1: **The posterior probability sharpens as more data are accumulated.** *Here we sampled data according to a Poisson distribution with $\lambda = 5$ (designated by the dotted line). Our samples were $w_{1:10} = \{2, 8, 5, 3, 5, 2, 5, 10, 6, 4\}$. We plotted the prior (eq. (4.6)) with $\alpha = 2$, $\beta = 1/7$) and the resulting posterior after collecting $N = 1$, then $N = 5$ and $N = 10$ points.*

As we will see, values for hyperparameters in conjugate priors take simple physical interpretation: they provide prior estimates of parameters of the likelihood and the prior sample size, called the conjugate prior pseudocount, used to arrive at this estimate. The latter concept of prior sample size is especially relevant to our discussion on the *prior strength*.

Beyond their obvious mathematical appeal, as we will see in subsequent chapters, their mathematical appeal provides direct computational advantage and, for this reason, they are commonly used in Bayesian applications provided likelihoods lend themselves to the formulation of conjugate priors.

4.3.3 Bayesian formulations in the exponential family

Likelihoods in the exponential family

A likelihood belonging to the exponential family of distributions is convenient as it suggests that the entire dataset can be reduced to point statistics. Under this circumstance, we can construct conjugate distributions which lead to dramatically improved computational tractability.

A distribution is said to belong to a *K-parameter exponential family* if the likelihood of a single data point can be written as

$$p(w|\theta) = f(w)g(\theta)\exp\left(\sum_{k=1}^{K}\phi_k(\theta)u_k(w)\right). \tag{4.9}$$

Example 4.3: The Normal belongs to the exponential family

Here we re-write a one-dimensional Normal to make it clear that it belongs to the 2-parameter exponential family.

To see this explicitly, we write

$$p(w_n|\theta = \{\mu, \sigma^2\}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(w_n-\mu)^2}{2\sigma^2}} \tag{4.10}$$

$$= \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\mu^2}{2\sigma^2}}\right) e^{\frac{w_n\mu}{\sigma^2} - \frac{w_n^2}{2\sigma^2}}. \tag{4.11}$$

Comparing eq. (4.11) to eq. (4.9) leads to the immediate identification: $f = 1$, $g = e^{-\frac{\mu^2}{2\sigma^2}}/\sqrt{2\pi\sigma^2}$, $K = 2$, $u_1 = w_n$, $\phi_1 = \mu/\sigma^2$, $u_2 = w_n^2$, $\phi_2 = -1/2\sigma^2$.

The likelihood for N data points, assumed iid, can be written as

$$p(w_{1:N}|\theta) = \left(\prod_n^N f(w_n)\right) g^N(\theta) \exp\left(\sum_{k=1}^K \phi_k(\theta) t_k(w_{1:N})\right) \tag{4.12}$$

where $t_k(w_{1:N}) = \sum_{n=1}^N u_k(w_n)$. The t_k's are called *sufficient statistics* as the collection of $t_{1:K}$ encapsulate all ways of combining the data to fully specify the likelihood. For example, the Exponential only has one sufficient statistics (the mean), while the Normal has two; the first and second moment or, equivalently, the mean and variance.

Priors in the exponential family

Given a likelihood of the form eq. (4.12), only if the prior has the form

$$p(\theta) \propto g^\eta(\theta) \exp\left(\sum_{k=1}^K \phi_k(\theta)\nu_k\right) \tag{4.13}$$

does the posterior regain the form of the prior

$$p(\theta|w_{1:N}) \propto g^{\eta+N}(\theta) \exp\left(\sum_{k=1}^K \phi_j(\theta)(t_k(w_{1:N}) + \nu_k)\right) \tag{4.14}$$

but with η in the prior substituted for $\eta + N$ in the posterior and similarly for ν_k and $\nu_k + t_k(w_{1:N})$.

Thus we see that, for conjugate priors, re-writing the likelihood and priors to make them take the explicit forms above, namely eq. (4.12) and eq. (4.13) respectively, means that one hyperparameter plays the role of a pseudocount (η) and the other plays the role of a pseudoestimate (ν_k) of the k^{th} sufficient statistic (t_k). We have such a pair of hyperparameters for each sufficient statistic appearing in the likelihood.

This identification of hyperparameters as pseudocounts and pseudoestimates was clear from the example we gave earlier on the Poisson likelihood and its corresponding Gamma prior. It would have been equally clear for the Beta-Bernoulli prior-likelihood conjugate pair had we re-written the likelihood and prior in the form of eq. (4.12) and eq. (4.13).

Informative priors for Normal likelihoods

Here add an example with the typical homework problem for the Normal. Tis example should continue with a graphical depiction in the next section and as an exerxise in the MCMC chapter.

Probably the most commonly used likelihood is the Normal or Gaussian likelihood for multiple reasons. First, it is computationally tractable and is completely defined by two parameters: the mean and variance. Secondly, it is an excellent physical approximation for physical experiments. The reason for this stems from the *central limit theorem*.

Loosely, this theorem states that if a random variable can be thought of as arising from the sum of multiply contributing realizations of random variables, and provided that each of those random variables is sampled from a

distribution with finite mean and variance, then the sum of those random variables will be distributed according to a Normal distribution. As experimental observations are often the output of multiple contribution factors, Normal distributions are ubiquitous across the Natural Sciences.

For iid experiments, a Normal likelihood takes the form

$$p(w_{1:N}|\mu,\tau^{-1}) = \text{Normal}(w_{1:N};\mu,\tau^{-1}) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} e^{-\tau\frac{(w_n-\mu)^2}{2}}. \tag{4.15}$$

where τ, termed the *precision parameter* is related to the variance by $\tau^{-1} = \sigma^2$.

For problems of dimension d, we often write

$$p(w_{1:N}|\mu,\mathbf{\Sigma}) = \prod_{n=1}^{N} \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} e^{-(w_n-\mu)^T\cdot\mathbf{\Sigma}^{-1}\cdot(w_n-\mu)} \tag{4.16}$$

where w_n is a vector of numbers associated to the n^{th} data point, μ is a mean vector of size d and $\mathbf{\Sigma}$ is a $d\times d$ covariance matrix. We turn to multi-dimensional problems in discussions on Gaussian processes in later chapters.

For the example of the Poisson likelihood seen earlier, we assumed that our goal was to generate a posterior over the only sufficient statistic of relevance to the problem, namely the mean.

As the Normal has two parameters, *i.e.* two sufficient statistics, we have three cases to consider. Either one of the two parameters is known (whilst our goal is then to generate a posterior over the other) or both parameters are unknown.

We start with the case of known mean and unknown variance and follow up immediately thereafter with the two subsequent cases.

Example 4.4: Conjugate prior to the Normal with known mean and unknown variance

The likelihood is

$$p(w_{1:N}|\mu,\tau^{-1}) \propto \tau^{N/2} exp\left(-\frac{\tau N\nu}{2}\right) \tag{4.17}$$

where we have defined $\nu = \frac{1}{N}\sum_n^N (w_n-\mu)^2$.

The conjugate prior to this likelihood, constructed using eq. (4.13), is the Gamma distribution

$$p(\tau|\alpha,\beta) \propto (\tau/\beta)^{\alpha-1}\exp(-\tau/\beta). \tag{4.18}$$

By comparing eq. (4.17) and eq. (4.18), we see that the hyperparameter α is used to compute the pseudocount (the pseudocount is $2(\alpha-1)$) and $\alpha\beta$ serves as the prior mean (pseudoestimate) of the precision.

The resulting posterior obtained by multiplying the likelihood and Gamma prior also assumes the form of a Gamma distribution

$$p(\tau|w_{1:N},\mu) \propto \tau^{N/2+\alpha-1} e^{-\tau\left(N\nu/2+\beta^{-1}\right)}. \tag{4.19}$$

Note that we place the mean on the left hand side of the conditional in the posterior, as it is a given.

Example 4.5: Conjugate prior to the Gaussian with unknown mean and known variance

We start again with the likelihood

$$p(w_{1:N}|\mu,\tau) \propto exp\left(-\tau\frac{\sum_n^N (w_n-\mu)^2}{2}\right). \tag{4.20}$$

The conjugate prior, constructed from eq. (4.13), is also a Normal

$$p(\mu|\xi,\psi) = \text{Normal}(\mu;\xi,\psi^{-1}). \tag{4.21}$$

The resulting posterior, obtained by multiplying prior and likelihood and completing the square to make the resulting product look like a Gaussian in μ, is

$$\mu|w_{1:N}, \tau^{-1} \sim \text{Normal}\left(\frac{\bar{w}N\tau + \xi\psi}{N\tau + \psi}, \frac{1}{N\tau + \psi}\right) \tag{4.22}$$

where $\bar{w} = N^{-1}\sum_n^N w_n$.

We immediately see that as N becomes very large, the Normal distribution mean reduced to $\bar{w}$ exactly as expected.

Example 4.6: Conjugate prior to the Gaussian with unknown mean and variance

We may start with eq. (4.22) and eq. (4.19) and write

$$\tau|\beta, \alpha \sim \text{Gamma}(\alpha, \beta) \tag{4.23}$$

$$\mu|\xi, \psi^{-1} \sim \text{Normal}(\xi, \psi^{-1}). \tag{4.24}$$

Assuming independent priors over μ and τ, we may write the joint prior as the product of both eq. (4.22) and eq. (4.19)

$$p(\mu, \tau|\xi, \psi, \beta, \alpha) = p(\mu|\xi, \psi)p(\tau|\beta, \alpha) \propto \tau^{\alpha-1}e^{-\frac{\tau}{\beta} - \psi\frac{(\mu-\xi)^2}{2}}. \tag{4.25}$$

When multiplying eq. (4.25) by a Normal likelihood, eq. (4.20), the posterior does not assume the form of the joint prior. For this reason, we say that the Gamma distribution is only conditionally conjugate, assuming known mean, and that the Normal distribution is *conditionally conjugate* assuming known variance.

Another choice of prior is to assume

$$\tau|\beta, \alpha \sim \text{Gamma}(\alpha, \beta) \tag{4.26}$$

$$\mu|\xi, \kappa_0\tau^{-1} \sim \text{Normal}(\xi, \kappa_0\tau^{-1}). \tag{4.27}$$

The joint prior then takes the form

$$p(\mu, \tau|\xi, \kappa_0, \beta, \alpha) = p(\mu|\xi, \kappa_0\tau^{-1})p(\tau|\beta, \alpha) \tag{4.28}$$

$$\propto \tau^{\alpha-1/2}e^{-\tau\left(\frac{\kappa_0(\mu-\xi)^2}{2} + \beta^{-1}\right)}. \tag{4.29}$$

Here, this prior, when multiplied by Normal likelihood, eq. (4.20), generates a posterior of the same form

$$p(\mu, \tau|w_{1:N}, \xi, \kappa_0, \beta, \alpha) \propto \tau^{N/2+\alpha-1/2}e^{-\tau\left(\frac{\kappa_0(\mu-\xi)^2}{2} + \beta^{-1} + \sum_n^N (w_n-\mu)^2\right)}. \tag{4.30}$$

Finding conjugate multivariate priors to likelihoods is generally not possible and of limited use. For this reason, we do not dwell on this topic here. However, the concept of *conditional conjugacy* will be heavily leveraged in later chapters.

Starting from $p(\mu, \tau|w_{1:N})$, where the hyperparameter dependency has been dropped for convenience, it is sometimes helpful to compute the marginal posterior $p(\mu|w_{1:N})$. That is, the posterior over the mean irrespective of the realizations over the variance.

As we will see from the example below, $p(\mu|w_{1:N})$ resembles a Cauchy distribution with heavy tails. This, in turn, raises an important question: how does the shape of $p(\mu|w_{1:N})$ accommodate outliers? Well, for a Cauchy-like distribution, increased probability density is re-assigned from the mode to the tails. How can we avoid outliers strongly disproportionately impacting the distribution in this way? This brings us to the treatment of outliers within the Bayesian paradigm and an implicit modeling choice we made in example 4.6 that we explore

in the example below.

Example 4.7: A Bayesian treatment of outliers

Suppose our goal is to estimate the posterior over the mean irrespective of the value assigned to the variance. In this case, we marginalize the posterior of eq. (4.30) over the variance

$$p(\mu|w_{1:N}) = \int_0^\infty d\tau p(\mu, \tau|w_{1:N}). \tag{4.31}$$

For simplicity, we drop all hyperparameter dependence on the right hand side of the conditional in $p(\mu|w_{1:N})$. The integral yields

$$p(\mu|w_{1:N}) \propto \left(1 + \beta\left(\frac{\kappa_0(\mu-\xi)^2}{2} + \sum_n^N (w_n - \mu)^2\right)\right)^{-(N/2+\alpha+1/2)}. \tag{4.32}$$

The posterior of eq. (4.32) has a single mode. This is evident by first taking the logarithm of the posterior and noting that its derivative with respect to μ is a linear equation supporting only one solution (*i.e.* one maximum) with respect to μ. As such, this posterior is globally impacted by the effect of its tails to accommodate outliers.

The question then becomes: how can we limit the impact of outliers to a local region of the posterior?

We first recognize that in constructing the conjugate prior over the mean and variance, we made important choices. One choice we made was to assume that the prior over the variance of each data point was the same. Yet, this logic runs counter to that anticipated in the presence of outliers where some data points must be attributed different, presumably larger, variances than others.

As such, we start by relaxing the constraint that all data points must have the same associated variance and write

$$p(\mu|w_{1:N}) \propto \int_0^\infty \prod_n d\tau_n \; p(w_n|\mu, \tau_n)p(\mu, \tau_n). \tag{4.33}$$

Under this circumstance,

$$p(\mu|w_{1:N}) \propto \prod_n \left(1 + \beta\left(\frac{\kappa_0(\mu-\xi)^2}{2} + (w_n - \mu)^2\right)\right)^{-(\alpha+1/2)}. \tag{4.34}$$

To see why this posterior is indeed multi-modal, we take the derivative with respect to μ of its logarithm. This derivative immediately yields a high order polynomial in μ suggesting multiple maxima and minima especially pronounced for large prior pseudocounts, κ_0.

This example illustrates that important limitation of maximizing either the posterior to obtain parameter estimates. It is clear from this example that multiple maxima suggest a more complex posterior structure worthy of reporting in full.

4.4 Graphical representations of Bayesian formulations

So far, we have considered measurement output, $w_{1:N}$, latent variables, labeled $r_{1:N}$, parameters of the measurement distribution, $\phi_{1:F}$, and other model parameters of the generative model that we call $\theta_{1:M}$ here. To be consistent with prior notation, we say that the total number of parameters, K, satisfies $K = F + M$.

In Bayesian analysis, we also discussed placing priors on all parameters to be determined. These could include $\phi_{1:F}$ and $\theta_{1:M}$ if all unknown. On these prior parameters, in turn, we may place hyperpriors and so forth.

The idea here is to graphically show the dependency of random variables on one another. To do so, we describe the ground rules of the graphical model: 1) every random variable of the model is shown; 2) each random variable

goes within a circle; 3) arrows pointing from one to another random variable indicate dependencies between the variables; finally 4) known random variables are shaded.

We begin with those random variables that are always known: observations. As we see below, these are shown with time, or measurement index, progressing from left to right. As we see on the right panel of this figure, these random variables can be collected in a plate whose running index coincides with those of the random variable.

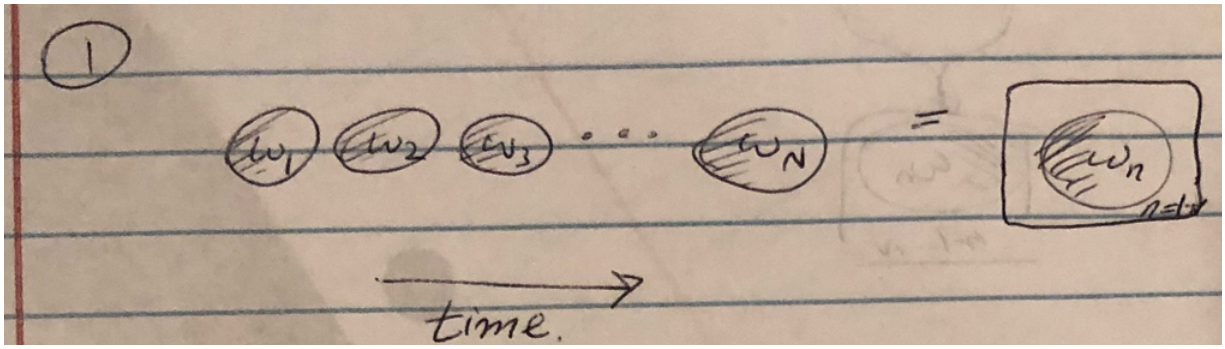


Figure 4.2

For simple cases, where do not have model latent variables, the observations themselves are directly informed by the parameters, $\theta_{1:M}$, of the generative model. In this case, and under the assumption that all parameters are unknown, we have an arrow pointing from those parameters shown in unshaded boxes to the measurements.

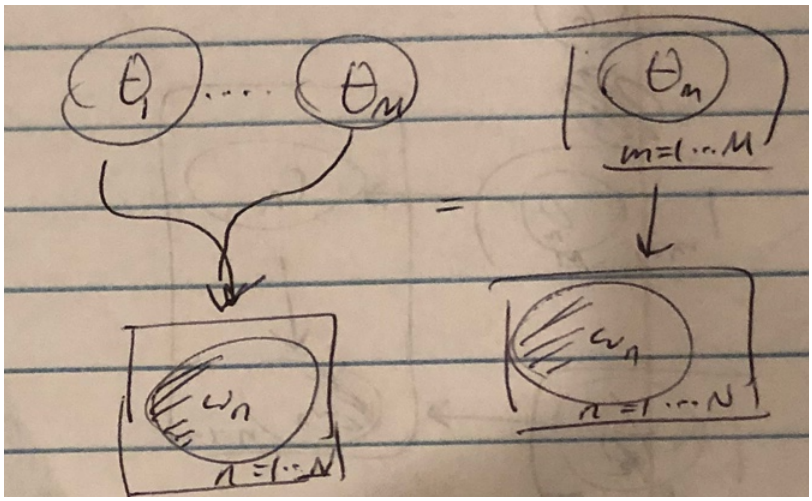


Figure 4.3

A concrete example of the above for iid Gaussian observations is provided below.

Example 4.8: Graphical model for iid Gaussian observations

Below we consider three cases shown side by side for data, $w_{1:N}$, generated from a normal distribution, Normal$(\mu, 1/\tau)$, The cases we consider are: μ and $1/\tau$ unknown; μ known and $1/\tau$ unknown; and μ unknown and $1/\tau$ known.

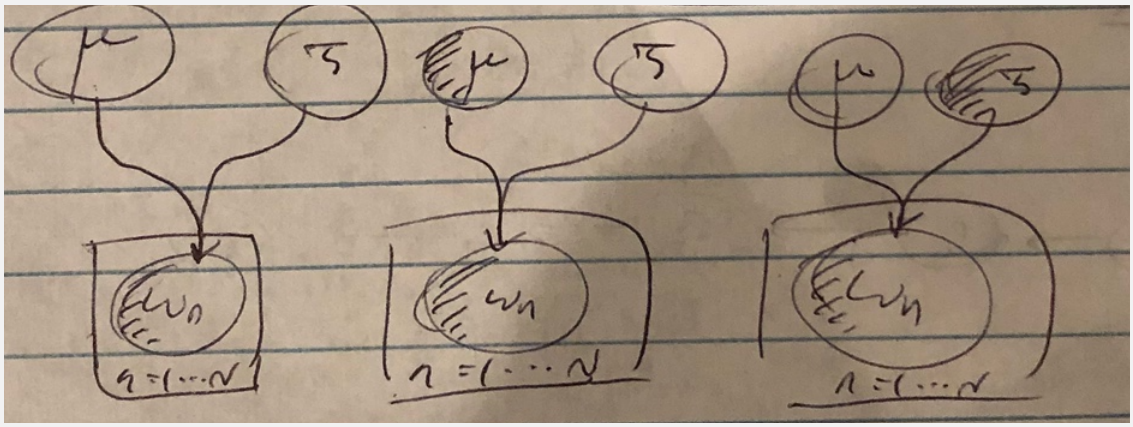


Figure 4.4

Typically, if we use conditionally conjugate priors for the mean, μ, and precision, τ, these priors are parameterized by hyperparameters. If these hyperparameters are pre-specified, they are not random variables and do not appear in circles called *nodes*. An example is provided below.

Example 4.9: Graphical model for iid Gaussian observations with explicit hyperparameters

Below we consider the case where data, $w_{1:N}$, is generated from a normal distribution, Normal$(\mu, 1/\tau)$. For simplicity, we only consider the case where μ and $1/\tau$ are unknown. If we insist on using conditionally conjugate priors, then the prior over μ is also Normal(ξ, ψ^{-1}) and that over τ is Gamma(α, β).

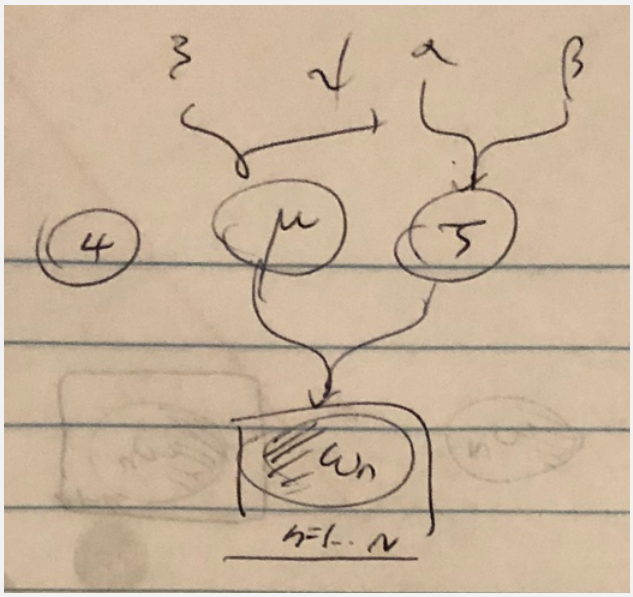


Figure 4.5

As we now introduce latent variables, we first focus on iid observations. In the presence of latent variables, the plates can be extended to include the latent variable as the index running over the latent variable and measurement are typically the same as show in the figure below.

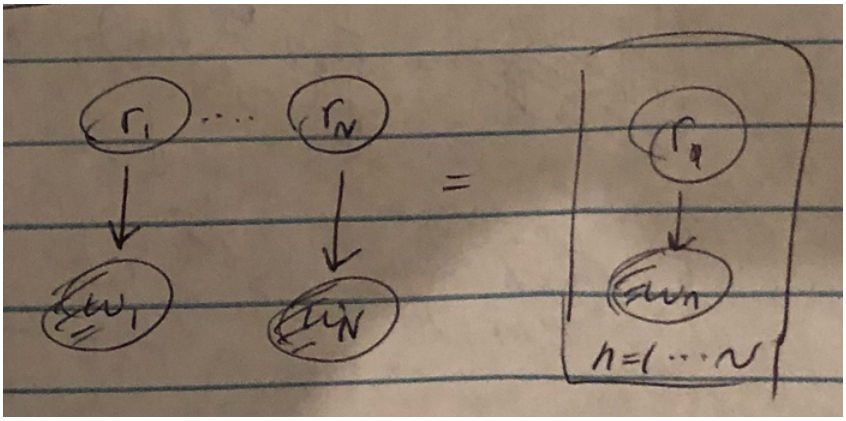


Figure 4.6

An example of this includes a mixture of Gaussian model, for a example a mixture of two Gaussians, where the latent variable r_n can be realized to one or two for each data point.

As we bring in latent variables, we begin discriminating between parameters of the measurement distribution and other parameters of the generative process. A general graphical representation for this scenario is provided in the subsequent figure.

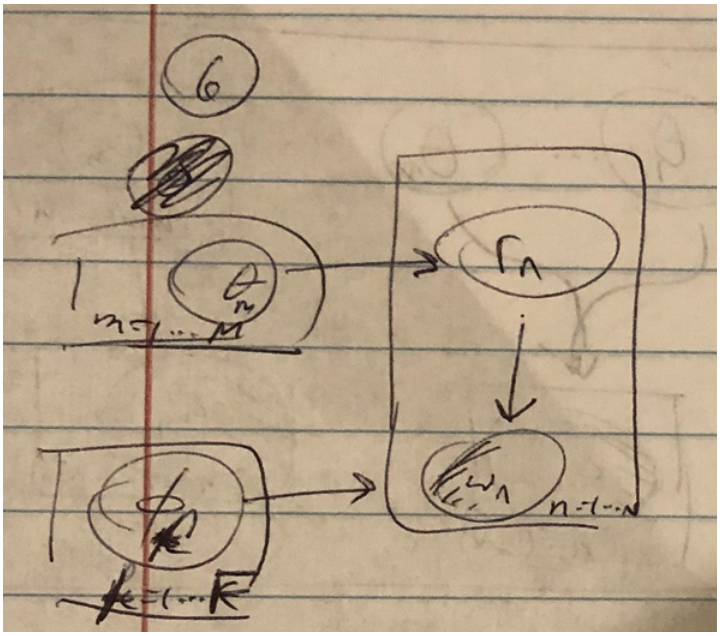


Figure 4.7

We immediately see that the above figure captures the following generative model structure

$$\begin{aligned} \theta_{1:M} &\sim \mathbb{P}(\cdot) \\ \phi_{1:F} &\sim \mathbb{P}(\cdot) \\ R_n &\sim \mathbb{P}(\theta_{1:M}) \\ W_n | r_n, \phi_{1:F} &\sim \mathbb{P}(r_n, \phi_{1:F}) \end{aligned}$$

A concrete example of this can be drawn from an example where samples are drawn from a mixture of Gaussians but subsequently corrupted by noise.

We would be remiss in our duty if we avoided dynamical generative, *i.e.* non-iid, examples. As we will delve into much greater detail on these models in later chapters, for now, we simply highlight the essentials.

We start with the simplest example of a dynamical model; one that satisfies the Markov property. The first clear observation here is that random variables generating the observation are not independent.

For example, in the simple case of diffusion, *i.e.* a Brownian process, the position at time level n depends on the position at previous times. An example of this is shown explicitly in the figure below.

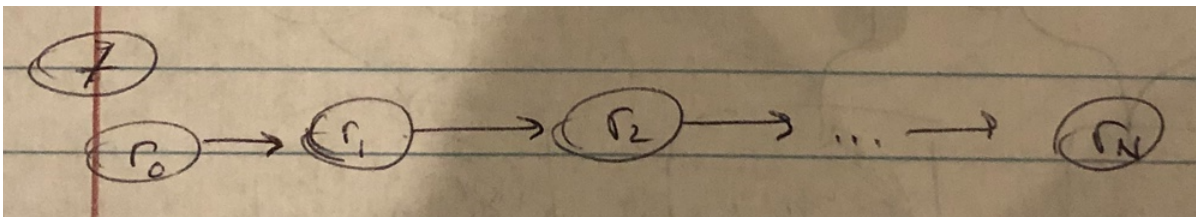


Figure 4.8

In this figure, r_0, denotes the initial condition. By modeling measurement noise explicitly, we introduce an observation coinciding with the latent variable at each time level.

To be explicit, we supplement this model with a prior on the measurement distribution parameters, $\phi_{1:F}$ and other parameters of the model, collectively termed $\theta_{1:M}$, including those parameters parametrizing the distribution over the initial condition, r_0, if unknown and those parameters dictating the kinetics as the system evolves across time levels.

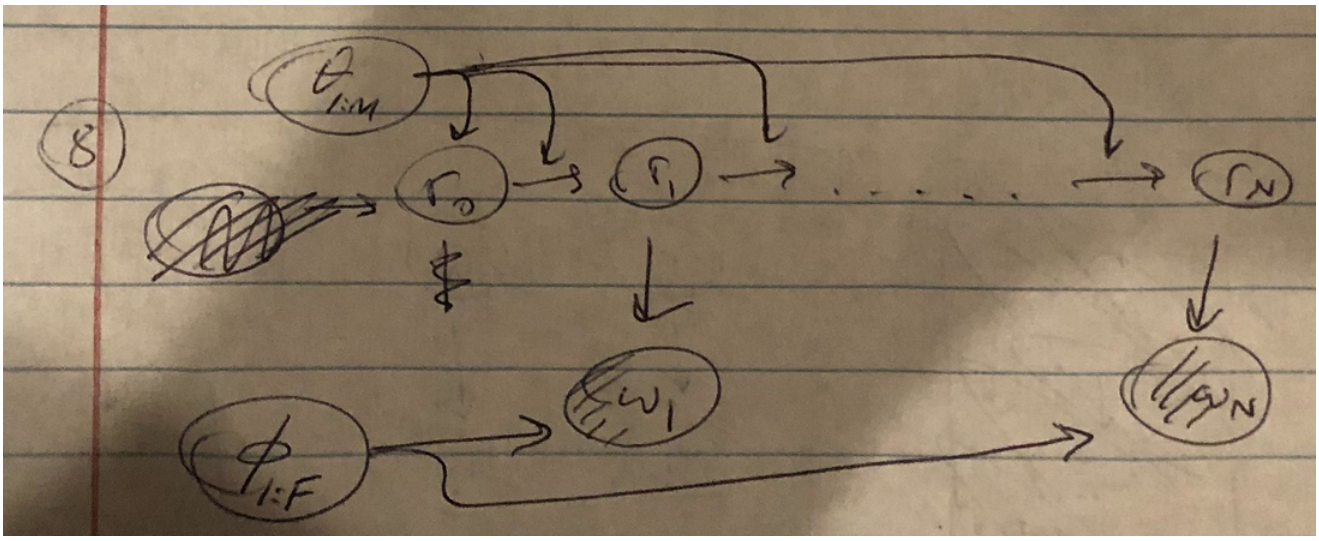


Figure 4.9

Finally, we end with a note on Markov blankets.

The Markov blanket of node, reflecting the random variable r, in a graphical representation includes the set of nodes that we denote $MB(r)$.

The idea is that, once conditioned on all members of its Markov blanket, a node is conditionally independent of all other nodes, r',

$$p(r|MB(r), r') = p(r|MB(r)). \tag{4.35}$$

As a consequence, this implies that for two nodes, r_i and r_j, with the same Markov blanket

$$p(r_i, r_j|MB(r_i)) = p(r_i|MB(r_i))p(r_j|MB(r_i)). \tag{4.36}$$

Which nodes should we include in the Markov blanket? The Markov blanket of a node r includes its parent nodes, its children and spouses (other parents of all of its children).

This seems reasonable as, clearly, knowing the state of its parent tells you something about r. Following similar logic, knowing something about the children tells you something about the parent. What is more, by considering the effect of the spouses on the children, we can eliminate factors influencing the children that are not due to the instantiation of r itself.

4.5 Bayesian model selection

4.5.1 The model selection problem

The topic of model selection is central to Bayesian analysis. In computing a Bayes' factor, note 4.2, we had assumed throughout that the model structure was unchanged and that we were solely comparing ratios of likelihoods (or posteriors) for different parameter values. Yet, all of this is under the assumption that the model structure is known.

Yet models may not be known *a priori*. For example, we can imagine a likelihood for iid outcomes distributed according to a mixture of Normal distributions with unknown number of mixture components. The likelihood with two mixture components will always fit the data at least as well as the simpler model. This is apparent from the fact that we are free to set the mixture weight of the second component to zero.

We can immediately intuit that as the model dimensionality increases, *i.e.* the number of mixture component grows, the fit to the data improves. Yet there is a flip side: higher-dimensional models are worse at predicting future data points.

Before discussing a Bayesian attempt at addressing model selection, we first illustrate the worsening predictive ability of complex models by discussing the predictive distribution.

Posterior or likelihood ratios are sensitive to model dimensionality. That is, neither posterior nor likelihood ratios are helpful in comparing models of different dimensionality.

Indeed, a model with a large number of parameters approaching the number of data points should fit the data almost exactly and yield a likelihood approaching its maximum theoretical value: unity. So why not make models arbitrarily complex?

Intuitively, we know that models should not be as complex as the data. In fact, the main objective behind the Natural Sciences is to reduce large data sets to key insights which scientists call models and computer scientists, in particular, call lossy compression. Otherwise, if the data is incompressible, then the model is, trivially, just the data.

More quantitatively, we know that a very large number of parameter numbers reduces our ability to make predictions about the realization of future data points. In other words, we say that a large number of parameters reduces the magnitude of the *p*redictive distribution, $p(w_{N+1}|w_{1:N}) = \int d\theta p(w_{N+1}|\theta)p(\theta|w_{1:N})$. This is because the model has, in some sense, overcommitted to the data set previously presented to it.

We see this explicitly with two examples below.

Example 4.10: Predictive distribution

Here we give the example of the value of the predictive distribution decreasing as the model complexity increases. We give the example of a Categorical likelihood with 3 possible outcomes such that each iid random variable sample is realized as 1 or 2 or 3.

In this case, the likelihood is

$$p(w_n|\boldsymbol{\pi}) = \pi_1^{\delta_1(w_n)}\pi_2^{\delta_2(w_n)}\pi_3^{\delta_3(w_n)}. \tag{4.37}$$

Below we introduce the prior conjugate to this 3-component Categorical distribution, the Dirichlet distribution, a distribution of key importance that we will revisit on multiple occasions in future chapters

$$p\left(\boldsymbol{\pi}|\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \alpha_3\}\right) = \frac{\Gamma(\alpha_1+\alpha_2+\alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)}\pi_1^{\alpha_1-1}\pi_2^{\alpha_2-1}\pi_3^{\alpha_3-1} \tag{4.38}$$

$$= \text{Dirichlet}(\boldsymbol{\pi}; \boldsymbol{\alpha}) \tag{4.39}$$

where $\{\alpha_1, \alpha_2, \alpha_3\}$ are hyperparameters (exceeding zero) and $\pi_1 + \pi_2 + \pi_3 = 1$.

As this prior is conjugate, the posterior also takes the form of a Dirichlet distribution with $\{\alpha_1, \alpha_2, \alpha_3\}$ in the prior updated to $\{\alpha_1 + n_1, \alpha_2 + n_2, \alpha_3 + n_3\}$ in the posterior where n_1 are the total number of times that outcome 1 was realized in $N = n_1 + n_2 + n_3$ experiments. Both n_2 and n_3 are similarly defined.

Next we compute the probability that w_{N+1} is realized to outcome 1 by computing the *predictive distribution* , $p(w_{N+1} = 1|w_{1:N}) = p(w_{N+1} = 1|w_{1:N}, \alpha)$. The predictive distribution reads

$$p(w_{N+1} = 1|w_{1:N}, \alpha) = \int_0^1 d\pi_3 \int_0^1 d\pi_2 \int_0^1 d\pi_1 \delta_1(\pi_1 + \pi_2 + \pi_3) p(w_{N+1} = 1|\pi) p(\pi|w_{1:N}, \alpha) . \tag{4.40}$$

Inserting eq. (4.37) and eq. (4.39) into eq. (4.40), we find

$$p(w_{N+1} = 1|w_{1:N}, \boldsymbol{\alpha}) = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3 + N)}{\Gamma(\alpha_1 + n_1)\Gamma(\alpha_2 + n_2)\Gamma(\alpha_3 + n_3)} \tag{4.41}$$

$$\int_0^1 d\pi_3 \int_0^1 d\pi_2 \int_0^1 d\pi_1 \delta_1(\pi_1 + \pi_2 + \pi_3) \pi_1^{\alpha_1 + n_1} \pi_2^{\alpha_2 + n_2 - 1} \pi_3^{\alpha_3 + n_3 - 1}. \tag{4.42}$$

By recognizing that the integral above is the same integral as would be required to normalize a Dirichlet distribution, we can immediately write down the integral above from the normalization of eq. (4.39). We find

$$p(w_{N+1} = 1|w_{1:N}, \boldsymbol{\alpha}) = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3 + N)}{\Gamma(\alpha_1 + n_1)\Gamma(\alpha_2 + n_2)\Gamma(\alpha_3 + n_3)} \frac{\Gamma(\alpha_1 + n_1 + 1)\Gamma(\alpha_2 + n_2)\Gamma(\alpha_3 + n_3)}{\Gamma(\alpha_1 + \alpha_2 + \alpha_3 + N + 1)} \tag{4.43}$$

$$= \frac{\alpha_1 + n_1}{\alpha_1 + \alpha_2 + \alpha_3 + N}. \tag{4.44}$$

In general, for the K-component Categorical likelihood (and thus K-component conjugate Dirichlet prior) we have

$$p(w_{N+1} = i|w_{1:N}, \alpha) = \frac{\alpha_n + n_n}{\sum_{k=1}^K \alpha_k + N}. \tag{4.45}$$

We immediately see that in comparing the predictive distribution of the two and three component models, we find $p(w_{N+1} = 1|w_{1:N}, \{\alpha_1, \alpha_2\}) > p(w_{N+1} = 1|w_{1:N}, \{\alpha_1, \alpha_2, \alpha_3\})$. That is, the less complex model (*i.e.* the 2-component Categoricallikelihood as compared to the 3-component Categorical likelihood) is, as predicted, strictly more predictive than than that of the more complex model.

Making the dimensionality of K explicit and writing $\alpha = \alpha_{1:K}$ for a K-component model, we would say

$$p(w_{N+1} = 1|w_{1:N}, \alpha_{1:K-1}) > p(w_{N+1} = 1|w_{1:N}, \alpha_{1:K}) \tag{4.46}$$

for integer $K \geq 2$.

Example 4.11: Bias-variance tradeoff

The bias-variance tradeoff or precision-accuracy tradeoff is a classic problem of regression where we ask what order polynomial is required to fit data. Here we suppose that the n^{th} data point is a function, f, of a control variable, x_n That is,

$$w_n = f(x_n) + \epsilon_n \tag{4.47}$$

where $\epsilon_n \sim \text{Normal}(0, \sigma^2)$.

If we assume that the function is linear, we write

$$w_n = \alpha + \beta x_n + \epsilon_n. \tag{4.48}$$

A more draconian assumption is to consider a constant function

$$w_n = \alpha + \epsilon_n. \tag{4.49}$$

In a problem at the end of the chapter, we ask the reader to show that the predictive distribution for the more complex model (*i.e.*, the linear function), $p(w_{N+1}|w_{1:N}, \alpha)$ where α are hyperparameters, has a broader variance than the simpler (constant) model. By contrast the bias, that is $\langle w_{N+1} \rangle - f(x_{N+1})$, is smaller for the linear function.

On the one hand, we want models complex enough to provide a good fit to the data. On the other hand, we don't want models so complex that the predictive distribution becomes vanishingly small when evaluated for reasonable values of w_{N+1}. This is the essence of *m*odel selection, one of the greatest modeling challenges.

What we do next is discuss some model selection criteria. One criterion, the Akaike information criteria which we will not be discussing here, is grounded in information theory and does not uniquely follow from Bayesian logic. Instead, here we discuss the Bayesian information criterion (BIC), sometimes called the Schwartz information criterion (SIC).

4.5.2 The Bayesian Information Criterion

The BIC tries to find a compromise between goodness of fit of the data while penalizing model complexity by identifying the most probable "model dimensionality" (which can refer to the number of parameters or components of a model) given the data.

For concreteness, here we call the dimensionality of the model K and think of the dimensionality as being related to the number of model parameters K, $\theta = \theta_{1:K}$. This strategy only works for models whose complexity can be shrunk or grown in dimensionality. Such models, whose parameters form a subset of the parameters of a larger model, are called *nested models*. For example a single Gaussian is a subset of a 2-component Gaussian mixture model. The 2-component mixture, in turn, is a subset of a mixture model with an even larger number of components.

In other words, we first compute $p(K|w_{1:N}) = \int d\theta_{1:K} p(\theta_{1:K}|w_{1:N})$ where the integral is over the entire range allowed by each parameter, and then find the K maximizing this posterior. Thus, we treat $p(K|w_{1:N})$ as a criterion, a Bayesian criterion, informing us of the optimal K.

Since, in general, we will not be able to perform the integral over all parameters θ exactly, we will, instead, compute this integral asymptotically in the large N limit.

To compute this posterior, we first define a likelihood, $p(w_{1:N}|\theta)$, describing N observations. For iid or, at the very least, weakly correlated observations, we imagine that the likelihood will take the form of a product over the likelihood for each data point. This argument, motivates re-writing the likelihood in the following suggestive form

$$p(w_{1:N}|\theta) \approx e^{N \log p_1(w_{1:N}|\theta)} \tag{4.50}$$

where $p_1(w_{1:N}|\theta)$ is interpreted as a likelihood per data point, $\sqrt[N]{p(w_{1:N}|\theta)}$. Strictly speaking, the linear N scaling preceding the logarithm of **??** only holds for iid measurements. The approximation breaks down entirely for Markov processes.

The posterior can now be written as

$$p(\theta, K|w_{1:N}) \propto e^{N \log p_1(w_{1:N}|\theta)} p(\theta, K). \tag{4.51}$$

Marginalizing over all parameter values, assuming K fixed, is accomplished as follows

$$p(K|w_{1:N}) = \int d\theta p(\theta, K|w_{1:N}). \tag{4.52}$$

The integral appearing in eq. (4.52) is, in general, impossible to take analytically and perhaps infeasible numerically depending on the exact form of the likelihood. However a large N suggests an asymptotic approximation.

The motivation here is simple. If N is large, the integral is dominated by that region of the integrand where $\log p_1(w_{1:N}|\theta)$ is at its maximum with respect to the set θ, which we call $\hat{\theta}$. By this argument, the likelihood can be approximated by its region neighboring $\theta = \hat{\theta}$. The prior over this narrow neighborhood is assumed roughly flat.

In light of this argument, we invoke *Laplace's method* and expand $\log p_1(w_{1:N}|\theta)$ around its maximum, $\hat{\theta}$, up to quadratic order. In other words, we write

$$\begin{aligned} p(K|w_{1:N}) &\propto e^{N \log p_1(w_{1:N}|\hat{\theta})} \int d\theta \; e^{-\frac{N}{2}(\theta-\hat{\theta})\cdot\nabla_\theta\nabla_\theta \log p_1(w_{1:N}|\hat{\theta})\cdot(\theta-\hat{\theta})} \\ &= e^{N \log p_1(w_{1:N}|\hat{\theta})} \frac{(2\pi/N)^{K/2}}{\sqrt{\det \; \nabla_\theta\nabla_\theta \log p_1(w_{1:N}|\hat{\theta})}}. \end{aligned} \tag{4.53}$$

The value of K maximizing this marginal posterior now coincides with the number of parameters we need. We have a choice to either maximize this marginal posterior or minimize the negative of its logarithm as both the logarithm of a function and the function itself are monotonic.

The choice of -2 multiplied by the logarithm of the posterior is historical and this following object is called the Bayesian information criterion (BIC)

$$BIC \equiv -2 \log p(K|w_{1:N}) = -2 \log p(w_{1:N}|\theta^*) + K \log N + \mathcal{O}(N^0). \tag{4.54}$$

The contribution from the prior to the BIC is asymptotically vanishing and contributes to the term of order N^0, which we ignore.

Note 4.5: Why the BIC works?

Why does the BIC penalize complex models or, equivalently, introducing new parameters? It appears we just integrated the plain old likelihood by first approximating it as a Gaussian, ignored the prior, and somehow got the likelihood to penalize model parameters. Not quite.

The key to this apparent conundrum as to why the likelihood penalizes model complexity lies in the non-trivial operation of integration itself.

The more model parameters we have, the more parameters we need to integrate over. Now, having many parameters, as we discussed earlier, improves fit to data and can make the likelihood approach unity. But this is only true of good parameter values. Poorly selected parameters, especially many multiple poorly selected parameter values, can dramatically reduce a likelihood. In other words, if we integrate over all parameter values that could have been possible with very high dimensional models, there are many more "ways" of reducing the posterior than there "ways" of increasing it.

The BIC beautifully captures the competition between wanting to improve the goodness of fit to the data (captured by the first term) without overfitting the model (captured by the second term). The second term, sometimes called a "complexity penalty", scales as $\log N$.

While our discussion above has remained abstract, below we give an example of the BIC as used in change point detection.

(Steve to Ioannis: I recognize the K in the example may not reflect the total number of parameters, as per our convention,...let me know your thoughts)

4.5.3 A case study in change-point detection

Example 4.12: Change-point detection

A creative use of the BIC lies in detecting change points in time series data . Change point algorithms locate points in the data where the statistics for a process generating the data change. For example, we may imagine data points being drawn from a Gaussian with some mean and variance for some initial portion of the trace followed by data points being drawn from a Gaussian with a different mean (but, for simplicity, only) the same variance for all subsequent time points.

Here we illustrate how model selection, and the BIC in particular, is explicitly applied to this change point detection problem with fixed but unknown standard deviation, assumed identical for all data points, and with a discretely changing mean (also unknown).

Model selection lies in the fact that each data points could be assumed to be sampled from a distribution with different means or that, in the other extreme, all data points are sampled from the same distribution.

We begin by writing down the likelihood

$$p(w_{1:N}|K,\sigma^2,\mu,j) = \prod_{k=0}^{K-1}\prod_{n=j_k}^{j_{k+1}-1}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(w_n-\mu_k)^2}{2\sigma^2}} \tag{4.55}$$

where K denotes the number of change-points – points where the mean of the signal changes occurring at locations $j = j_{0:K}$. To be precise, since the standard deviation is also a parameter to be determined, we have $K+1$ total parameters here.

The model maximizing this likelihood places a change-point at every step ($w_n = \mu_k$ for every n). That is, $p(w_{1:N}|K,\sigma,\mu_{1:N},j)$ peaks when there is a change point in each data point.

To avoid overfitting, and since we are still largely ignorant of the correct values for σ and μ, we integrate over all allowed values for σ and μ and, following BIC logic for sufficiently large N, assume that whichever prior we choose varies slowly (and is thus essentially constant) in the region over which the likelihood peaks. In other words,

$$p(K,j|w_{1:N}) \propto \int d\sigma d\mu\ p(w_{1:N}|K,\sigma^2,\mu,j). \tag{4.56}$$

To do these integrals, we first re-write the likelihood in eq. (4.56) as

$$p(w_{1:N}|K,\sigma^2,\mu,j) = \frac{1}{(2\pi\sigma^2)^{N/2}}e^{-\sum_{k=0}^{K-1}\sum_{n=j_k}^{j_{k+1}-1}\frac{(w_n-\mu_k)^2}{2\sigma^2}}. \tag{4.57}$$

We then plug eq. (4.57) into eq. (4.56) and do the Gaussian integrals required to marginalize over μ. This returns

$$p(K,j|w_{1:N}) \propto \int d\sigma \frac{1}{\sigma^N}\left(\sigma^K e^{-\frac{1}{2\sigma^2}S}\right) \tag{4.58}$$

where $S \equiv n_0\hat{\sigma}_0^2 + \cdots + n_{K-1}\hat{\sigma}_{K-1}^2$ and

$$\hat{\sigma}_k^2 \equiv \frac{1}{n_k}\sum_{n=j_k}^{j_{k+1}-1} w_n^2 - \frac{1}{n_k^2}\left(\sum_{n=j_k}^{j_{k+1}-1} w_n\right)^2 \tag{4.59}$$

where n_k counts the number of points contained in the k^{th} step ($j_{k+1} - j_k$).

We must still integrate over σ. To do this final integral, we recognize that the argument of eq. (4.58) can be transformed, by variable substitution, into a Gamma function. Thus, the subsequent integral over σ yields the

following marginal posterior

$$p(K, j|w_{1:N}) \propto \int d\sigma d\mu \; p(w_{1:N}|K, \sigma^2, \mu, j)$$
$$= \frac{\sqrt{2\pi}^{-(N-K)}}{n_0^{1/2} \cdots n_K^{1/2}} \cdot \frac{1}{2} \cdot \left(\frac{S}{2}\right)^{-\frac{(N-K-1)}{2}} \cdot \left(\frac{N-K-3}{2}\right)! \tag{4.60}$$

It is now, no longer obvious that the resulting $p(K, j|w_{1:N})$ of eq. (4.60) is peaked for K approaching N. Indeed taking the further simplifying assumptions that all n_k are large we recover a simplified form for a Gaussian likelihood BIC sometimes seen in the literature

$$BIC = -2 \log p(K, j|w_{1:N}) = N \log S + K \log N + \mathcal{O}(N^0) + \text{const} \tag{4.61}$$

where the constants, const, capture all terms independent of K (but that may otherwise depend on N).

Fig. (4.10) shows the detection of change points in synthetic data and illustrates just how sensitive the BIC is to the correct choice of likelihood. To address this sensitivity, BIC's have, for example, been tailored to detect change points with time correlated noise.

The change point algorithm used to generate the figure is provided below.

Algorithm 4.1: A greedy BIC change-point algorithm for Gaussian likelihoods

We start from a time trace that we assume satisfies the likelihood of eq. (4.55). We use the BIC of eq. (4.61) to find the change points using a greedy method.
Start by assuming there are no change-points and compute the BIC.

- Introduce an additional change point at all possible locations where there are no current change points.
- If the addition of a change point at any location does not reduce the BIC, as compared to the BIC with one fewer change points, stop.
- Otherwise, if a change point does reduce the BIC, find the best location (the one that minimizes the BIC) and fix the change point there (hence the "greedy" nature of the algorithm) and return to the first step.

There are small issues one may quibble about regarding the BIC. Here's one: what if the model that informs our likelihood can only logically increase the number of parameters in groups of 3 say (imagine introducing a new Gaussian in a mixture with its own weight, mean and variance) while the BIC is perfectly happy providing a maximum in K that is not a multiple of 3? Should we then round K up or down to the closest multiple of 3?

The latter is a small issue. There is a much more important challenges that remain. What do we do after having selected a K? Do we go back to square one and, armed with K (rounded up or down to the closest multiple that we need), construct a likelihood of the appropriate size and proceed onward with constructing a posterior?

Somehow this feels *ad hoc* as models of different sizes cannot "cross-talk". Perhaps for an audience knowledgeable of introductory quantum mechanics the analogy to quantum superposition and interference might be helpful here. A probability amplitude which appears as a linear superposition of two-states generates a probability (the absolute square of the amplitude) with four terms. The cross-terms matter. In fact, the cross-terms are what distinguish classical from quantum mechanics and this distinction revolutionized 20th century science.

By close analogy, one's optimal choice of parameters in a model of complexity K is not independent of one's choice of optimal parameters for a model of complexity $K + 1$. In other words, model parameters are not independent across models. Yet, it is precisely this assumption that is made in treating parameter maximization and model dimensionality estimation as two separate, artificially disjoint, steps.

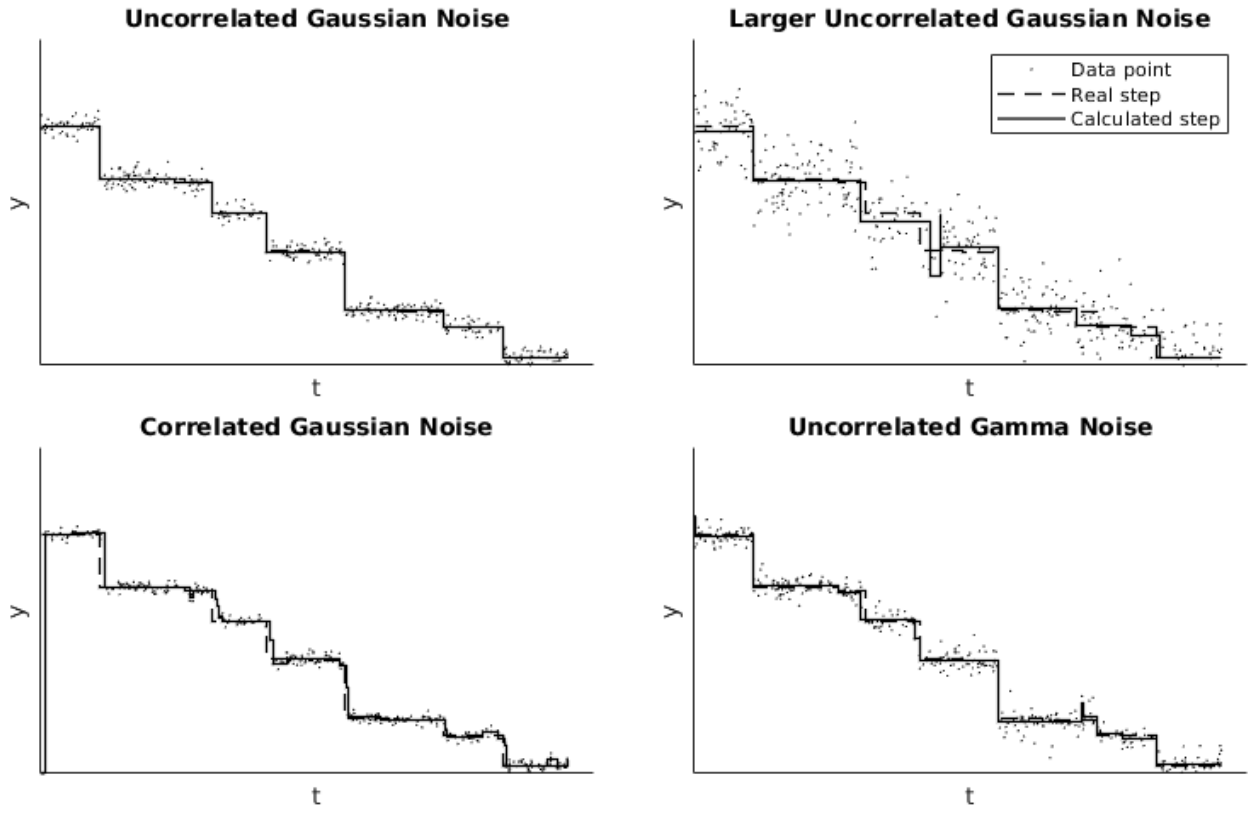


Figure 4.10: The BIC locates correct steps when the noise statistics are well characterized. **a)** *Our control. We generated synthetic steps (black line) and added noise (white, decorrelated) with the same standard deviation for each data point. We used a greedy algorithm to identify and compare models according to eq.* (4.61) *and identify the correct step locations (red line) from the noisy time trace (blue).* **b)** *Here we use a different, incorrect, likelihood that does not adequately represent the process that we used to generate the synthetic data. That is, we correctly assumed that the noise was white and decorrelated but also, incorrectly, assumed that we knew and fixed σ (and therefore did not integrate over σ in eq.* (4.56)*). We underestimated σ by 12%. Naturally, we overfit (red) the true signal (black). Green shows the change point algorithm re-run using the correct noise magnitude.* **c)** *Here we use the BIC from eq.* (4.61) *whose likelihood assumes no noise correlation. However, we generated a signal (black) to which we added correlated noise [by first assigning white noise, ϵ_t, to each data point and then computing a new correlated noise, $\tilde{\epsilon}_t$, at time t from $\tilde{\epsilon}_t = 0.7\epsilon_t + 0.1\epsilon_{t-1} + 0.1\epsilon_{t-2} + 0.1\epsilon_{t-3}$]. As expected, the model that the BIC now selects (red) interprets as signal some of the correlated noise from the synthetic data.*

This deeper model selection problem is elegantly resolved by Bayesian nonparametrics (BNPs) by turning parameter dimensionality estimation into a (normal) Bayesian parameter estimation problem. We turn to BNPs in later chapters.

4.6 Exercise problems

Exercise 4.1

A physical system is pumped with light at every time interval T. Each M molecule within the physical system has an equal probability, P_{exc}, of being excited. If it is, a molecule emits a photon after an exponentially distributed time. As soon as the first photon strikes the detector, no new photon can be recorded until the next T. How are the photon arrivals distributed?

Exercise 4.2

Compute the predictive distribution, $p(w_{N+1} = 1|w_{1:N})$, for a Poisson likelihood with a conjugate Γ prior.

Exercise 4.3

Imagine an experiment where event arrival times are recorded: t_1, t_2, etc... This experiment reveals that the waiting time between these events is exponentially distributed, i.e. $\lambda e^{-\lambda t}$ where λ, as we saw in class, is an event's rate of arrival. What is the waiting time distribution for 2 events (in other words, what is the waiting time for $T = t_1 + t_2$)? Generalize this to N events. We now have $P(T|N)$. But we are interested in $P(N|T)$. Use Bayes' theorem, and the fact that $P(N) = 1$ (i.e. the probability of achieving N events regardless of the time), to show that $P(N|T)$ is Poisson distributed, $(\lambda T)^N e^{-\lambda T}/N!$.

Exercise 4.4

In this chapter, we explored bias-variance tradeoff, also called the precision-accuracy tradeoff. We reconsider the n^{th} data point, w_n, given by a function, f, of a control variable, x_n. That is, $w_n = f(x_n) + \epsilon_n$ where $\epsilon_n \sim$ Normal$(0, \sigma^2)$.
Generate data according to a quintic function $f(x_n)$. Compute the predictive distribution $p(w_{N+1}|w_{1:N}, \boldsymbol{\alpha})$ assuming a constant, linear and quadratic dependency of f on x_n. Here $\boldsymbol{\alpha}$ are hyperparameters of any reasonable prior you chose.
Show that the variance of the predictive distribution is largest assuming quadratic f and smallest assuming constant f. Show that the bias, as defined in example 4.11, is greatest for the constant and smallest for the quadratic function.

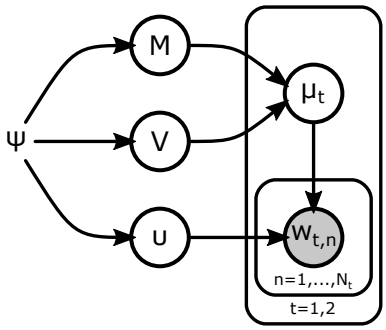
Exercise 4.5

Change-point algorithm
Implement the BIC-based change-point method for the simple case of a Gaussian likelihood with identical standard deviation assumed for all data points. In other words, begin by generating synthetic data assuming discrete steps in your data where you have pre-specified by hand the means of those steps in the signal, $\mu_{1:K}$, and σ.
Then, implement the change-point method in order to learn the means, $\mu_{1:K}$, and locations, j, of your steps. Compare your parameter estimates from your method to the theoretical values you used to generate your data.

Exercise 4.6: Graphical representations

In this problem, you will identify a graphical model.

1. List exhaustively every variable shown in the graphical model on the right.
2. Sort the variables you identified into three groups: random variables with known values, random variables with unknown values, and variables that are not random.


M
V
u
ψ
μt
wt,n
n=1,...,Nt
t=1,2

Chapter 5

Computational inference

By the end of this chapter, we will have presented
- *The Metropolis-Hastings and Gibbs algorithms*
- *Schemes for Bayesian computations*
- *Strategies to sample arbitrary distributions*

As we have seen in earlier chapters, the solution of an inverse problem through Bayesian inference relies on posterior probability distributions and often requires revealing their shape or, at a more quantitative level, evaluating integrals with respect to them. Except in rare cases, however, the posterior distributions involved in our problems cannot be derived analytically. Here we describe computational methods that relax this limitation.

5.1 The fundamentals of MCMC

5.1.1 Monte Carlo methods

Monte Carlo (MC) methods provide a wide class of algorithms that can be used as alternative approaches for problems that can be reframed as probabilistic when analytic approaches are impossible. MC methods apply broadly, even beyond Bayesian problems. We briefly summarize their use below.

Note 5.1: What is a Monte Carlo method?

A Monte Carlo method describes a computational approach that generates and uses pseudo-random numbers. Such methods are mainly used to solve three wide types of problems:

1. optimization: where we seek to identify conditions that lead to optimal results according to a predefined criterion,
2. integration: where we seek to compute an integral of some predefined quantity,
3. sampling: where we seek to obtain samples of some quantity that follows given predefined statistics.

Such problems arise commonly in the Sciences and MC methods are employed extensively, especially to investigate cases involving complex physical systems that consist of numerous interacting components, such as molecules, cells, or agents of any kind. Two common features of the problems MC handles efficiently, namely complex interactions and high dimensionality, make alternative approaches, *i.e.* fully deterministic approaches, practically infeasible.

Despite being used mostly to simulate specific experiments or to test scientific theories, as we will see in this chapter, MC methods can also be used to solve inverse problems, *i.e.* to sample from distributions such as posteriors.

MC methods are used to solve problems attaining a probabilistic interpretation and we have already seen some examples: Box-Muller in note 1.8 and the Gillespie simulation in algorithm 2.2, amongst others. In concrete terms, with MC we wish to characterize how a random variable R is distributed. Generally, R may be univariate or multivariate, it may contain continuous or discrete components, and may gather *model parameters* and/or

auxiliary variables. As usual, in the Bayesian context we are mostly interested in unveiling a posterior probability distribution of R or its corresponding posterior probability density $p(r|w)$ for appropriate observations that we denote with the variable w.

To keep the notation simple, throughout this chapter, we denote the posterior $p(r|w)$ simply as $\pi(r)$ and, since the methods we will describe can be applied also in cases where $\pi(r)$ may not necessarily be a posterior, but any probability density, we call $\pi(r)$ the *target density*. For clarity, we denote with Π the target's distribution. In this setup, our targeted random variable is denoted with $R \sim \Pi$ and its density is denoted with $\pi(r)$.

The way MC methods work is as follows. First, we simulate a sequence $r^{(1)}, r^{(2)}, \dots, r^{(J)}$ of iid samples of R. In other words, we obtain realizations $r^{(j)} \sim \Pi$. As we will see, we can generally perform such sampling even when an analytic formula for $\pi(r)$ is unavailable. Subsequently, we may use the sampled sequence of $r^{(j)}$ to estimate expectations

$$\langle V \rangle = \int_r dr\, g(r)\pi(r) \tag{5.1}$$

of any quantity $V = g(R)$ that is a function of R. Of course, if V contains discrete components, the integral in eq. (5.1) reduces to an appropriate sum. Although we will not explicitly discriminate between integrals and sums, the description that follows remains valid even in this case.

Since $\langle V \rangle$ in eq. (5.1) is the mean of a transformed random variable, $V = g(R)$, we use $v^{(j)} = g(r^{(j)})$ to compute samples of this random variable and subsequently use them to evaluate $\langle V \rangle$. Formally, this means that with MC we consider the approximation

$$\underbrace{\langle V \rangle}_{\substack{\text{desired}\\ \text{expectation}}} \approx \frac{1}{J}\sum_{j=1}^{J} v^{(j)} = \underbrace{\frac{1}{J}\sum_{j=1}^{J} g(r^{(j)})}_{\substack{\text{sample}\\ \text{mean}}} \tag{5.2}$$

i.e. the sample mean of $g(r^{(j)})$ approximates the expectation $\langle V \rangle$ that we seek. In a similar manner, when the full distribution of V or a histogram of V are desired, this may be constructed by binning the samples $v^{(j)}$ instead of using them to compute a point statistic as in eq. (5.2).

Example 5.1: MC for the conjugate Normal-Gamma model

Suppose we have observations w_n, for $n = 1, \dots, N$, that are normally distributed and our task is to estimate the center and the spread of the underlying distribution. For this, we can consider a parametrization of the Normal distribution by mean μ and precision τ. On μ and τ, we place the common Normal-Gamma prior.

In statistical notation, the Bayesian model, which is fully conjugate, reads

$$\tau \sim \text{Gamma}(\alpha, \beta) \tag{5.3}$$

$$\mu|\tau \sim \text{Normal}\left(\xi, \frac{1}{\phi\tau}\right) \tag{5.4}$$

$$w_n|\mu,\tau \sim \text{Normal}\left(\mu, \frac{1}{\tau}\right), \qquad n = 1, \dots, N \tag{5.5}$$

where α, β, ξ, ϕ are hyper-parameters which, in this example, we assume to have some known values.

Within this set-up, the model is described by the random variable $r = \{\mu, \tau\}$, and the associated posterior is $\pi(r) = p(\mu, \tau|w_{1:N})$. As we have seen in **??**, we may compute samples $r^{(j)} = \{\mu^{(j)}, \tau^{(j)}\}$ from $\pi(r)$ by first generating

$$\tau^{(j)} \sim \text{Gamma}\left(\alpha + \frac{N}{2}, \beta + \frac{N}{2}\left(\bar{s} + \frac{\phi}{\phi + N}(\bar{w} - \xi)^2\right)\right) \tag{5.6}$$

and then generating

$$\mu^{(j)} \sim \text{Normal}\left(\frac{\phi\xi + N\bar{w}}{\phi + N}, \frac{1}{(\phi + N)\tau^{(j)}}\right) \tag{5.7}$$

where $\bar{w} = \frac{1}{N}\sum_{n=1}^{N} w_n$ and $\bar{s} = \frac{1}{N}\sum_{n=1}^{N}(w_n - \bar{w})^2$ are, effectively, constants since they rely only upon the observations $w_{1:N}$.

Once a sufficient number, J, of samples $r^{(j)}$ are generated through eqs. (5.6) and (5.7), we can then apply the MC technique described above. For example, some cases of interest might depend on whether

- we wish to visualize the posterior distribution of μ and τ. In this case, we may use $g(r) = (\mu, \tau)$ with random variable $V = (\mu, \tau)$. Here, a simple 2D histogram of $v^{(j)}$ would reveal the entire shape of our posterior $p(\mu, \tau | w_{1:N})$.
- we would like to compute the expectation of μ for which we would have to use another functional form for $g(r)$. In particular, since

$$\text{Mean of } \mu = \langle \mu \rangle = \int_r dr\, \mu\pi(r),$$

we would have to use $g(r) = \mu$, and so the above expectation would be approximated by

$$\langle \mu \rangle \approx \frac{1}{J}\sum_{j=1}^{J} \mu^{(j)}. \tag{5.8}$$

- we are interested in the variance of μ, which corresponds to the expectation

$$\text{Variance of } \mu = \langle (\mu - \langle \mu \rangle)^2 \rangle = \int_r dr\ (\mu - \langle \mu \rangle)^2\, \pi(r).$$

In this case, the function we would need to use is $g(r) = (\mu - \langle \mu \rangle)^2$, and to obtain $\langle \mu \rangle$, in the first place, we would need to use eq. (5.8). This choice leads to the approximation

$$\langle (\mu - \langle \mu \rangle)^2 \rangle \approx \frac{1}{J}\sum_{j=1}^{J} \left(\mu^{(j)} - \langle \mu \rangle\right)^2.$$

As an illustrative example, we consider a total of $N = 10$ observations, w_n, generated through Normal$(5, 1)$. Such observations and the generating distribution is shown in fig. 5.1A. Suppose parameters, *i.e.* mean and variance, of the generating distribution are unknown and we need to infer them from the available observations. For this, we analyze $w_{1:N}$ as described above.

For simplicity, we may make some vague choices for the hyper-parameters $\alpha = 2, \beta = 1, \xi = 2, \phi = 1$ and generate a total of $J = 120$ samples according to eqs. (5.6) and (5.7) that ensure reasonably flat priors over a broad range. These are shown in fig. 5.1B. For sake of a comparison, we also show the prior of μ, τ.

To obtain a more qualitative picture, fig. 5.1C compares both prior $p(\mu, \tau)$ and posterior $p(\mu, \tau | w_{1:N})$ densities, after the samples $r^{(j)}$ have been binned to produce a 2D histogram.

According to the posterior $p(\mu, \tau | w_{1:N})$, an estimate of the center of our observations is offered by the expectation of the mean μ, which is given by

$$\langle \mu \rangle \approx \frac{1}{J}\sum_{j=1}^{J} \mu^{(j)} = 4.50,$$

while, an estimate of the spread is offered by the expectation of the standard deviation $\sigma = \sqrt{1/\tau}$, which is given by

$$\langle \sigma \rangle = \left\langle \sqrt{\frac{1}{\tau}} \right\rangle \approx \frac{1}{J}\sum_{j=1}^{J} \sqrt{\frac{1}{\tau^{(j)}}} = 1.04.$$

We can see in example 5.1 that, once a sequence of samples $r^{(j)}$ is computed, the remaining MC choices, such

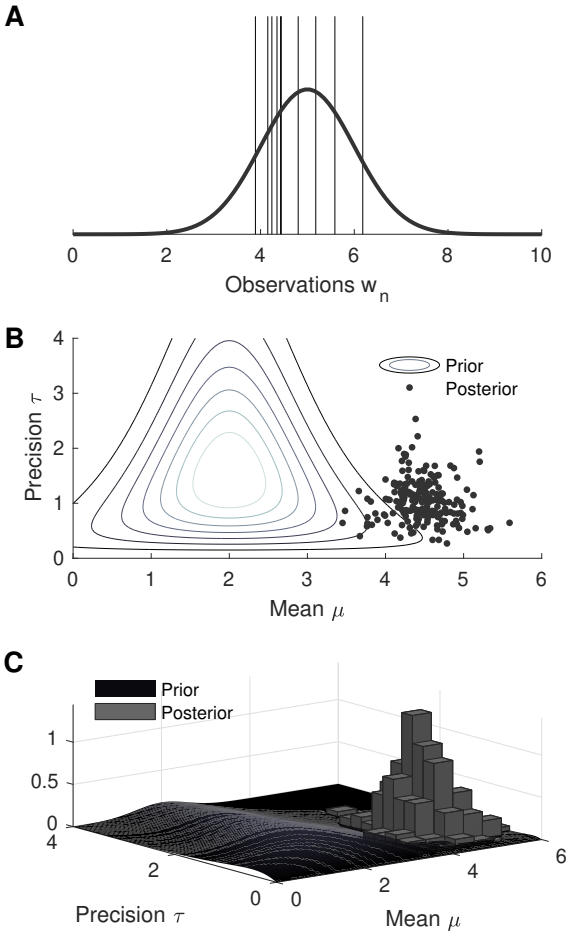


Figure 5.1: Illustration of example 5.1. Panel A: observations $w_{1:N}$ (vertical lines) and generating distribution (solid line). Panel B: prior probability distribution $p(\mu, \tau)$ (contours) and MC samples $\{\mu^{(j)}, \tau^{(j)}\}$ from the posterior $p(\mu, \tau | w_{1:N})$ (dots). Panel C: comparison of prior $p(\mu, \tau)$ (surface) and posterior $p(\mu, \tau | w_{1:N})$ (histogram).

as selecting the functional form of $g(r)$, are straightforward. However, unlike this example, where a computational recipe for sampling the random variable of interest R was already available, *e.g.* eqs. (5.6) and (5.7), generally obtaining samples from an arbitrary target $\pi(r)$ is a demanding task.

For this reason, we focus the rest of this chapter on the development of generic sampling schemes that can be used for the simulation of the sequence $r^{(j)}$ in the first place. In this chapter, we will go into greater detail and describe a particular class of MC methods, namely Markov chain Monte Carlo (MCMC), that are particularly suited for the solution of inference problems.

Note 5.2: What is a Markov chain Monte Carlo method?

MC methods are commonly used for simulation and data analysis to characterize complicated probability distributions. The idea is that, even if we are unable to evaluate the distribution of interest analytically, we can generate random samples from this distribution and, as we will see shortly, use these samples to derive relevant quantities, such as point statistics and histograms.

MCMC algorithms, however, do not generate independent samples from the distribution of interest, but rather generate Markov chains of samples. This is the reason for its appellation as *Markov chain* Monte Carlo (MCMC). The main distinction between MC and MCMC is that the former generates independent samples; while, as will see shortly, the latter generates dependent samples.

5.1.2 Markov chain Monte Carlo methods

As it is suggested by their name, MCMC simulations are performed in a sequential manner. Namely, each new sample $r^{(j+1)}$ of our targeted random variable $R \sim \pi$ is generated in a stochastic way that *depends* on the immediate previous sample $r^{(j)}$. Formally, this means that every MCMC method relies on successive samples

$$r^{(0)} \to r^{(1)} \to r^{(2)} \to \cdots \to r^{(j)} \to r^{(j+1)} \to \cdots \to r^{(J)}$$

in a Markov chain. So, our focus from now on shifts to the description of appropriate implementation schemes for the transitions $\cdots \to r^{(j-1)} \to r^{(j)} \to r^{(j+1)} \to \cdots$. Similar to MC, the MCMC samples $r^{(j)}$ carry the same statistical value as the target distribution itself π. So, even when we are unable to evaluate $\pi(r)$ directly, for example because it does not have a tractable formula or we are unable to evaluate it directly, we can still utilize the sequence of samples $r^{(j)}$ for the same purpose. However, unlike MC samples that are uncorrelated, MCMC samples *are correlated.* For this reason, to use approximations like eq. (5.2) we need to ensure that our Markov chain satisfies certain requirements that we highlight below.

Before delving into finer details, we first lay down some prerequisites. Similar to the ideas we encountered in chapter 2, Markov chains are best studied as sequences of random variables

$$R^{(0)} \to R^{(1)} \to R^{(2)} \to \cdots \to R^{(j)} \to R^{(j+1)} \to \cdots \to R^{(J)}$$

and from this perspective, each MCMC sample $r^{(j)}$ is a realization of a random variable $R^{(j)}$ in this sequence. To simplify our presentation from now on, instead of $R^{(j)}$ and $R^{(j+1)}$, we will use the more intuitive R^{old} and R^{new} to denote a random variance in this chain and its immediate successor. Similarly, we will be using r^{old} and r^{new} to denote realized values of these variables.

We will designate a sample's value r as *feasible* when it is allowed by the target, *i.e.* $\pi(r) > 0$. We will designate a chain as *ergodic* when its samples, in the long run, approach any feasible value, *i.e.* as the chain grows larger, formally at the limit $J \to \infty$, samples $r^{(j)}$ pass or come arbitrarily close to any feasible r.

Similar to any Markov chain, the transition rules leading from one MCMC sample to the next one take the form of conditional probability densities $p(r^{\text{new}}|r^{\text{old}})$. These densities quantify how likely a current sample r^{old} is to move to the next one r^{new} and generally say little about how the chain got into r^{old} which, due to the Markov property, is unimportant. To avoid any notational confusion, we will use $T_{r^{\text{old}}}(r^{\text{new}})$ to denote the transition densities of our MCMC chain and $\mathbb{T}_{r^{\text{old}}}$ to denote their corresponding distributions, *i.e.* in the convention we will be using from now on $T_{r^{(j)}}(r^{(j+1)}) = p(r^{(j+1)}|r^{(j)})$ and $R^{(j+1)}|r^{(j)} \sim \mathbb{T}_{r^{(j)}}$. Additionally, as we silently did

so far, we will only occasionally distinguish explicitly between random variables $R^{\text{old}}, R^{\text{new}}$ and their realizations $r^{\text{old}}, r^{\text{new}}$, as what we mean will be clear from the context.

Before we move on, we also introduce two important notions. A probability density $\pi^*(r)$ is termed *stationary* or *invariant* under the transitions $T_{r^{\text{old}}}(r^{\text{new}})$, when it satisfies the *full balance* condition

$$\pi^*(r^{\text{new}}) = \int_{r^{\text{old}}} dr^{\text{old}}\, T_{r^{\text{old}}}(r^{\text{new}})\pi^*(r^{\text{old}}).$$

Similarly, a density $\pi^*(r)$ is termed *reversible* under $T_{r^{\text{old}}}(r^{\text{new}})$, when it satisfies the *detailed balance* condition

$$T_{r^{\text{old}}}(r^{\text{new}})\pi^*(r^{\text{old}}) = T_{r^{\text{new}}}(r^{\text{old}})\pi^*(r^{\text{new}}).$$

It is easy to show that detailed balance implies full balance. For example, provided $r^{\text{old}} \sim \pi^*(r)$, we get

$$\begin{aligned} p(r^{\text{new}}) &= \int_{r^{\text{old}}} dr^{\text{old}}\, p(r^{\text{new}}, r^{\text{old}}) = \int_{r^{\text{old}}} dr^{\text{old}}\, p(r^{\text{new}}|r^{\text{old}})p(r^{\text{old}}) \\ &= \int_{r^{\text{old}}} dr^{\text{old}}\, T_{r^{\text{old}}}(r^{\text{new}})\pi^*(r^{\text{old}}) = \int_{r^{\text{old}}} dr^{\text{old}}\, T_{r^{\text{new}}}(r^{\text{old}})\pi^*(r^{\text{new}}) \\ &= \pi^*(r^{\text{new}}) \int_{r^{\text{old}}} dr^{\text{old}}\, T_{r^{\text{new}}}(r^{\text{old}}) \\ &= \pi^*(r^{\text{new}}). \end{aligned}$$

Consequently, detailed balance is a *stronger* condition than full balance.

With all prerequisites established, we are now ready to see the requirements of a valid MCMC method.

Note 5.3: MCMC requirements

To generate MCMC samples $r^{(j)}$ from a target $R \sim \mathbb{\Pi}$, we compute a Markov chain

$$r^{(0)} \to \cdots \to r^{(j-1)} \to r^{(j)} \to r^{(j+1)} \to \cdots$$

such that: (i) it has a stationary distribution; (ii) this distribution is unique; and (iii) it coincides with our target.

These requirements are fairly general and, in practice, we ensure them by developing MCMC schemes that generate chains under more restrictive conditions. Namely

- *The chain must pass from a feasible sample.* This is met as long as the initial or any subsequent sample is a feasible one.
- *The chain must be ergodic.* This is met as long as $T_{r^{\text{old}}}(r^{\text{new}})$ allows for successive transitions between every pair of feasible values.
- *The chain must have a stationary density.* This is met as long as $T_{r^{\text{old}}}(r^{\text{new}})$ is in full balance with some density $\pi^*(r)$. Besides ensuring requirement (i), this condition combined with initialization and ergodicity, ensures also requirement (ii).
- *The stationary density must coincide with the target.* This is met as long as $T_{r^{\text{old}}}(r^{\text{new}})$ is in detailed balance with our target $\pi(r)$. This condition ensures requirement (iii).

These conditions are termed: *feasibility, irreducibility, invariance, and reversibility,* respectively.

In practice, since detailed balance implies full balance, it is *only* necessary to ensure that the MCMC scheme satisfy the feasibility, irreducibility and reversibility conditions. Generally, feasibility poses no difficulties since the initial sample $r^{(0)}$ in an MCMC scheme is specified in advance independently of the subsequent transitions and its feasibility can be readily verified. The same is also true of the irreducibility condition which, aside from cases involving pathological transition rules $T_{r^{\text{old}}}(r^{\text{new}})$, is typically met or at least can be readily verified. However, reversibility is usually difficult to satisfy for arbitrary targets $\pi(r)$. So, in the next sections, we describe the most common MCMC schemes, or *samplers* as they are most commonly termed, for which reversibility is ensured by construction.

5.2 Basic MCMC samplers

In this section we present two general strategies of obtaining MCMC samplers. Namely, those that rely on the *Metropolis-Hastings algorithm* and those that rely on *Gibbs sampling*. In essence, both strategies utilize the same theoretical foundations; but, since their implementation differs, we discuss them separately.

As we will see, in both strategies, the target density $\pi(r)$ need not be normalized. So, we may apply them even if the target is specified only up to a multiplicative constant, *i.e.* a factor that *does not* depend on r, as usually is the case in Bayesian applications. To distinguish a target that may not be normalized, we will write $\bar{\pi}(r)$. Given an unnormalized target $\bar{\pi}(r)$, we can readily recover the corresponding normalized one by $\pi(r) = \bar{\pi}(r) / \int_{r'} dr' \, \bar{\pi}(r')$, and for this reason, we use $\pi(r)$ and $\bar{\pi}(r)$ interchangeably.

Note 5.4: Unnormalized targets

When we operate on an unnormalized target, $\bar{\pi}(r)$, it is critical to recall that this target is associated with a probability distribution $\mathbb{\Pi}$. Accordingly, we must ensure that our $\bar{\pi}(r)$ passes a simple sanity check, namely, that it is $\bar{\pi}(r)$ is *normalizable*. This holds as long as

$$0 < \int_r dr \, \bar{\pi}(r) < \infty$$

with *both inequalities* being important, otherwise $\bar{\pi}(r)$ is meaningless alongside any results derived from it.

5.2.1 Metropolis-Hastings family of samplers

Samplers in this family can be used to generate MCMC samples $r^{(j)}$ from virtually any random variable $R \sim \mathbb{\Pi}$. These include univariate or multivariate ones.

Metropolis-Hastings sampler

Given a target $\bar{\pi}(r)$, to begin a Metropolis-Hastings sampler, a choice for the initial sample $r^{(0)}$ needs to be made. This can be achieved either by assigning a fixed value or sampling a value from a specified probability distribution which need not necessarily be $\mathbb{\Pi}$. In the later case, the chosen distribution must exclude infeasible values. In any case, irrespective of how $r^{(0)}$ is computed, the sampler remains valid as long as $\bar{\pi}(r^{(0)}) > 0$. Next, $r^{(0)}$ is used to generate a subsequent sample, $r^{(1)}$, which, in turn, is used to generate $r^{(2)}$, and so on.

To advance the MCMC chain, the Metropolis-Hastings sampler uses a *proposal* density $p(r^{\text{prop}}|r^{\text{old}})$. We denote the proposal density with $Q_{r^{\text{old}}}(r^{\text{prop}}) = p(r^{\text{prop}}|r^{\text{old}})$ and the associated distribution with $\mathbb{Q}_{r^{\text{old}}}$. The proposal density can be almost arbitrary since its only requirements are that:

- the simulation of random variables $r^{\text{prop}}|r^{\text{old}} \sim \mathbb{Q}_{r^{\text{old}}}$ is possible,
- the simulation of random variables $r^{\text{prop}}|r^{\text{old}} \sim \mathbb{Q}_{r^{\text{old}}}$ allows the generation of any feasible value.

To advance from an existing sample r^{old} to the next one r^{new} in the MCMC chain, the Metropolis-Hastings sampler first generates a proposal sample by $r^{\text{prop}}|r^{\text{old}} \sim \mathbb{Q}_{r^{\text{old}}}$ and then conducts a test, which is based on the ratio

$$A_{r^{\text{old}}}(r^{\text{prop}}) = \underbrace{\frac{\bar{\pi}(r^{\text{prop}})}{\bar{\pi}(r^{\text{old}})}}_{\text{target}} \underbrace{\frac{Q_{r^{\text{prop}}}(r^{\text{old}})}{Q_{r^{\text{old}}}(r^{\text{prop}})}}_{\text{proposal}}, \tag{5.9}$$

to check whether r^{prop} is *retained or discarded*. Most commonly, for the test, a random number $u \sim \text{Uniform}_{[0,1]}$ is generated, then:

- If $u \leq A_{r^{\text{old}}}(r^{\text{prop}})$, the proposal is accepted and so the new sample is $r^{\text{new}} = r^{\text{prop}}$.
- If $u > A_{r^{\text{old}}}(r^{\text{prop}})$, the proposal is rejected and so the new sample is $r^{\text{new}} = r^{\text{old}}$.

Once r^{new} is obtained this way, either through acceptance or rejection of r^{prop}, the Metropolis-Hastings sampler applies the same process again, and again, until we reach a desired number of samples. In algorithm 5.1 we summarize a computational implementation.

Algorithm 5.1: Metropolis-Hastings sampler for arbitrary targets

Given a target $\bar{\pi}(r)$, a proposal $Q_{r^{\text{old}}}(r^{\text{prop}})$, and a feasible initial sample $r^{(0)}$, the Metropolis-Hastings sampler proceeds as follows:
For each j from 1 to J:

- Generate a proposal $r^{\text{prop}} \sim \mathbb{Q}_{r^{(j-1)}}$.
- Compute the acceptance ratio $A_{r^{(j-1)}}(r^{\text{prop}})$.
- Generate $u \sim \text{Uniform}_{[0,1]}$.
- If $u < A_{r^{(j-1)}}(r^{\text{prop}})$; set $r^{(j)} = r^{\text{prop}}$, else set $r^{(j)} = r^{(j-1)}$.

We emphasize that in every iteration, whenever the proposal r^{prop} is rejected, it is necessary to maintain r^{old}. In other words, in a correct implementation of the Metropolis-Hastings sampler, following a rejection, we *must* use $r^{\text{new}} = r^{\text{old}}$. As we will see in the next section, if we neglect such repetition, the sampler will fail to provide correct results.

Example 5.2: Two Metropolis-Hastings schemes for the truncated Normal distribution

Consider a random variable R distributed according to a Normal distribution with mean μ and variance σ^2 *truncated* below 0. That is, R has a probability density given by

$$\pi(r) \propto \bar{\pi}(r) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r-\mu)^2}{2\sigma^2}\right), & r \geq 0 \\ 0, & r < 0 \end{cases}.$$

It might be surprising, but despite its simplicity, there are no standard ways of simulating this target. So, to draw samples $r^{(j)}$ from $\pi(r)$ we may develop a Metropolis-Hastings sampler.

One convenient choice of the proposal is offered by $\mathbb{Q}_{r^{\text{old}}} = \text{Normal}(r^{\text{old}}, \lambda^2)$; that is, a Normal with mean on the previous sample r^{old} and a pre-set variance λ^2. This choice of proposal leads to the acceptance ratio

$$A_{r^{\text{old}}}(r^{\text{prop}}) = \begin{cases} \exp\left(\frac{(r^{\text{old}}-\mu)^2-(r^{\text{prop}}-\mu)^2}{2\sigma^2}\right), & r^{\text{prop}} \geq 0 \\ 0, & r^{\text{prop}} < 0 \end{cases}.$$

Of course, when implementing algorithm 5.1 we need not consider cases with $x^{\text{old}} < 0$ in the acceptance ratio, since r^{old} is ensured to be positive already. If this was not true, an acceptance of an infeasible value in the previous iteration or an infeasible initialization must had occurred, none of which is allowed.

One possible drawback of the Normal proposal used above is that it may often propose negative values r^{prop}, especially if μ is close to 0, and so it may lead to considerable rejections. A different choice that avoids such unnecessary rejections is offered by a $\mathbb{Q}_{r^{\text{old}}} = \text{Gamma}(\alpha, r^{\text{old}}/\alpha)$; that is, a Gamma distribution with mean on the previous sample r^{old} and a pre-set shape α, which is ensured to propose only positive values. This choice leads to the acceptance ratio

$$A_{r^{\text{old}}}(r^{\text{prop}}) = \begin{cases} \exp\left(\frac{(r^{\text{old}}-\mu)^2-(r^{\text{prop}}-\mu)^2}{2\sigma^2} + \alpha\left(\frac{r^{\text{prop}}}{r^{\text{old}}} - \frac{r^{\text{old}}}{r^{\text{prop}}}\right)\right)\left(\frac{r^{\text{prop}}}{r^{\text{old}}}\right)^{1-2\alpha}, & r^{\text{prop}} \geq 0 \\ 0, & r^{\text{prop}} < 0 \end{cases}.$$

Figure 5.2 illustrates both proposal choices. The target distribution is obtained with $\mu = 1$, $\sigma = 1$ and the proposals are implemented with $\lambda^2 = 0.2$ and $\alpha = 4$. For both cases, sampling starts from $r^{(0)} = 1$ and continued for a total of $J = 10^5$ samples. As can be seen, independently of the choice of the proposals, the target distribution is well characterized by the generated MCMC chains.

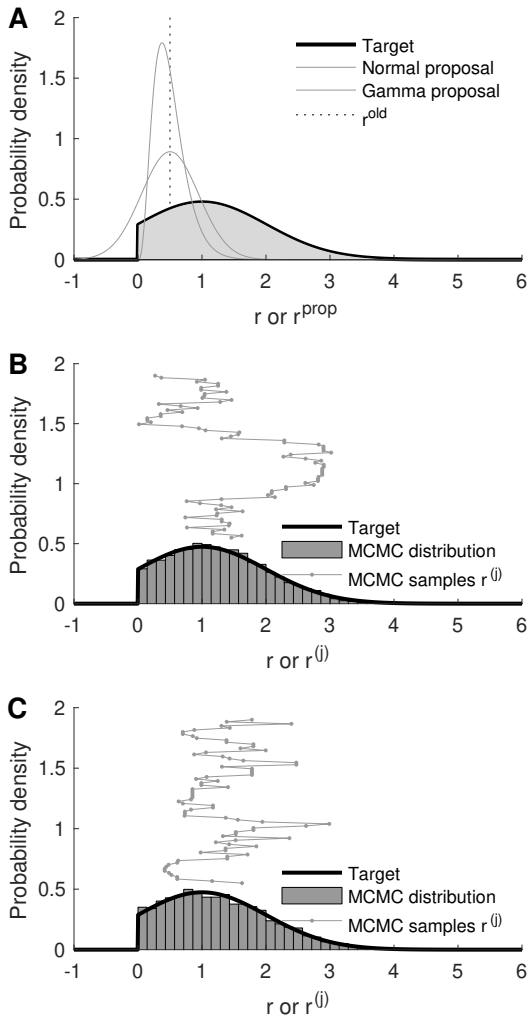


Figure 5.2: Illustration of example 5.2. Panel A: truncated Normal target $\pi(r)$ with Normal and Gamma proposals $Q_{r^{\text{old}}}(r^{\text{prop}})$. Panel B: MCMC approximation of $\pi(r)$ produced by Normal proposals. Panel C: MCMC approximation of $\pi(r)$ produced by Gamma proposals. Panels B and C also show parts of the generated Markov chains $\cdots \to r^{(j-1)} \to r^{(j)} \to r^{(j+1)} \to \cdots$. For clarity, only small segments of the two chains are shown.

Note 5.5: A sanity check for acceptance ratios

As we mention in earlier chapters, probability densities have units. Accordingly, our target $\bar{\pi}(r)$ and proposal $Q_{r^{\text{old}}}(r^{\text{prop}})$ also have units; however, since for convenience we often drop normalization constants, such units may not necessarily be equal to the inverse units of r. In any case, normalized or not, because eq. (5.9) consists of ratios of similar densities, the resulting $A_{r^{\text{old}}}(r^{\text{prop}})$ remains unit-less irrespectively of missing constants.

Why the sampler works?*

Given a target $\pi(r)$ and a proposal of choice $Q_{r^{\text{old}}}(r^{\text{prop}})$, the Metropolis-Hastings sampler ensures that the MCMC chain generated visits each feasible r with a frequency proportional to $\pi(r)$. This means that the sampler, as it passes from sample to sample balances out two competing effects: (i) how often the chain stays in each r and (ii) once left, how soon the chain returns back to r. As we can deduce from the acceptance test, the ratio $A_{r^{\text{old}}}(r^{\text{prop}})$ determines whether the sampler accepts a proposal, so the chain moves to a different value $r^{\text{new}} = r^{\text{prop}} \neq r^{\text{old}} \neq r^{\text{prop}}$, or rejects the proposal and the chain retains its value $r^{\text{new}} = r^{\text{old}}$. Accordingly, $A_{r^{\text{old}}}(r^{\text{prop}})$ plays an important role in settling both (i) and (ii).

Essentially, the acceptance test is a bookkeeping mechanism in which $A_{r^{\text{old}}}(r^{\text{prop}})$ keeps track of the balance between (i) and (ii). Intuitively, in eq. (5.9), the first term $\bar{\pi}(r^{\text{prop}})/\bar{\pi}(r^{\text{old}})$ balances the flow out of r^{old} and into r^{prop}; while, the second term $Q_{r^{\text{prop}}}(r^{\text{old}})/Q_{r^{\text{old}}}(r^{\text{prop}})$ balances the flow of getting from r^{old} to r^{prop} and getting back from r^{prop} to r^{old}.

To advance a deeper understanding of the sampler, we explain now the bookkeeping mechanism in detail. Specifically, recalling note 5.3, we will verify the *reversibility* condition. For this, we will explicitly show the chain's transition rules $T_{r^{\text{old}}}(r^{\text{new}})$ and show that these rules are in *detailed balance* with the target $\pi(r)$.

Transition rules First, to find out what the transition rules $T_{r^{\text{old}}}(r^{\text{new}})$ are, we need to consider all possibilities for r^{prop}. In other words, we need to complete over R^{prop}. Formally, the steps are

$$\begin{aligned} T_{r^{\text{old}}}(r^{\text{new}}) &= p(r^{\text{new}}|r^{\text{old}}) \\ &= \int_{r^{\text{prop}}} dr^{\text{prop}}\, p(r^{\text{new}}, r^{\text{prop}}|r^{\text{old}}) \\ &= \int_{r^{\text{prop}}} dr^{\text{prop}}\, p(r^{\text{new}}|r^{\text{prop}}, r^{\text{old}}) p(r^{\text{prop}}|r^{\text{old}}) \\ &= \int_{r^{\text{prop}}} dr^{\text{prop}}\, p(r^{\text{new}}|r^{\text{prop}}, r^{\text{old}}) Q_{r^{\text{old}}}(r^{\text{prop}}). \end{aligned}$$

To proceed any further, we must determine $p(r^{\text{new}}|r^{\text{prop}}, r^{\text{old}})$ which depends on the outcome of the acceptance test. Due to the way the test is performed, the probability of success $\alpha_{r^{\text{old}}}(r^{\text{prop}})$ is a function of both $r^{\text{old}}, r^{\text{prop}}$ and is given by

$$\alpha_{r^{\text{old}}}(r^{\text{prop}}) = \text{Probability of } \left(u \leq A_{r^{\text{old}}}(r^{\text{prop}})\right) = \min\left(1, A_{r^{\text{old}}}(r^{\text{prop}})\right).$$

Similarly, the probability of failure is also a function of both $r^{\text{old}}, r^{\text{prop}}$ and is given by $1 - \alpha_{r^{\text{old}}}(r^{\text{prop}})$.

Note 5.6: Acceptance probabilities

In exercise 5.6, we will prove an important identity

$$\pi(r)\alpha_r\left(r'\right) Q_r\left(r'\right) = \pi(r')\alpha_{r'}\left(r\right) Q_{r'}\left(r\right)$$

which is valid for any feasible r and r'. We will use this identity to show that the Metropolis-Hastings sampler meets the reversibility condition.

*This is an advanced topic and could be skipped on a first reading.

Since, once r^{prop} is drawn, there are only two options for r^{new}, namely either r^{prop} or r^{old}, we reach

$$p(r^{\text{new}}|r^{\text{prop}}, r^{\text{old}}) = \underbrace{\alpha_{r^{\text{old}}}(r^{\text{prop}})\delta_{r^{\text{prop}}}(r^{\text{new}})}_{\text{acceptance}} + \underbrace{(1 - \alpha_{r^{\text{old}}}(r^{\text{prop}}))\,\delta_{r^{\text{old}}}(r^{\text{new}})}_{\text{rejection}}.$$

Now, with $p(r^{\text{new}}|r^{\text{prop}}, r^{\text{old}})$ derived explicitly, we may continue with the transition rule

$$T_{r^{\text{old}}}(r^{\text{new}}) = \alpha_{r^{\text{old}}}(r^{\text{new}})Q_{r^{\text{old}}}(r^{\text{new}}) + \delta_{r^{\text{old}}}(r^{\text{new}}) \int_{r^{\text{prop}}} dr^{\text{prop}}\ (1 - \alpha_{r^{\text{old}}}(r^{\text{prop}}))\, Q_{r^{\text{old}}}(r^{\text{prop}}).$$

This is the most general form we can derive without consider specific choices of $\pi(r)$ and $Q_{r^{\text{old}}}(r^{\text{prop}})$. Next, we show that this form is already sufficient to verify the reversibility condition.

Balance condition The form of $T_{r^{\text{old}}}(r^{\text{new}})$ we have derived can be used to verify that the transition rules are in detailed balance with the target. In particular, in view of note 5.6, we obtain

$$\begin{aligned}
T_{r^{\text{old}}}(r^{\text{new}})\pi(r^{\text{old}}) &= \pi(r^{\text{old}})\alpha_{r^{\text{old}}}(r^{\text{new}})Q_{r^{\text{old}}}(r^{\text{new}}) \\
&\quad + \pi(r^{\text{old}})\delta_{r^{\text{old}}}(r^{\text{new}}) \int_{r^{\text{prop}}} dr^{\text{prop}}\ (1 - \alpha_{r^{\text{old}}}(r^{\text{prop}}))\, Q_{r^{\text{old}}}(r^{\text{prop}}) \\
&= \pi(r^{\text{new}})\alpha_{r^{\text{new}}}(r^{\text{old}})Q_{r^{\text{new}}}(r^{\text{old}}) \\
&\quad + \pi(r^{\text{new}})\delta_{r^{\text{new}}}(r^{\text{old}}) \int_{r^{\text{prop}}} dr^{\text{prop}}\ \left(1 - \alpha_{r^{\text{new}}}(r^{\text{old}})\right) Q_{r^{\text{new}}}(r^{\text{prop}}) \\
&= T_{r^{\text{new}}}(r^{\text{old}})\pi(r^{\text{new}}).
\end{aligned}$$

Sampling of posterior targets

The Metropolis-Hastings sampler can be applied to yield samples from any target $\bar{\pi}(r)$. In particular, such a target may be the posterior of a Bayesian model. For this special case, the acceptance ratio in eq. (5.9) becomes

$$A_{r^{\text{old}}}(r^{\text{prop}}) = \underbrace{\frac{p(w|r^{\text{prop}})}{p(w|r^{\text{old}})}}_{\text{likelihoods}} \underbrace{\frac{p(r^{\text{prop}})}{p(r^{\text{old}})}}_{\text{priors}} \underbrace{\frac{Q_{r^{\text{prop}}}(r^{\text{old}})}{Q_{r^{\text{old}}}(r^{\text{prop}})}}_{\text{proposals}} \tag{5.10}$$

for which we have used Bayes' theorem, $\pi(r) = p(r|w) \propto p(w|r)p(r)$, to factorize the posterior in the product of the likelihood $p(w|r)$ and the prior $p(r)$. Of course, following note 5.4, to do so, we need to ensure that the evidence $p(w)$ is non-zero. In other words, before our computations, we need to ensure that the data supplied *can* be generated by the model at hand.

Note 5.7: Initialization of the Metropolis-Hastings sampler

For Bayesian applications with complex or computational expensive likelihoods $p(w|r)$, checking the feasibility of the starting value $r^{(0)}$ may be demanding. For such cases, a convenient solution is to sample $r^{(0)}$ directly from the prior. As the priors are most often described by hierarchal or generative models, their sampling typically can be carried out directly.

Often, individual terms in the acceptance ratio, especially when the supplied datasets are large, become extremely small. For such cases, to avoid numerical underflow, the acceptance test can be performed in log-space. For example, we can directly compute the log-ratio

$$L_{r^{\text{old}}}(r^{\text{prop}}) = \log \frac{p(w|r^{\text{prop}})}{p(w|r^{\text{old}})} + \log \frac{p(r^{\text{prop}})}{p(r^{\text{old}})} + \log \frac{Q_{r^{\text{prop}}}(r^{\text{old}})}{Q_{r^{\text{old}}}(r^{\text{prop}})}$$

and, after generating $u \sim \text{Uniform}_{[0,1]}$, perform the acceptance test as:

- If $\log u \leq L_{r^{\text{old}}}(r^{\text{prop}})$, the proposal is accepted and so the new sample is $r^{\text{new}} = r^{\text{prop}}$.
- If $\log u > L_{r^{\text{old}}}(r^{\text{prop}})$, the proposal is rejected and so the new sample is $r^{\text{new}} = r^{\text{old}}$.

Example 5.3: MCMC for the conjugate Normal-Gamma model

To demonstrate the application of the Metropolis-Hastings sampler in the Bayesian context, consider the same setting as in example 5.1. Specifically, observations $w_{1:N}$ are normally distributed and our task is to estimate the center and the spread of the generating distribution. As before, parametrizing the underlying normal by mean μ and precision τ and placing a Normal-Gamma prior on them, we arrive at the model

$$\begin{aligned} \tau &\sim \text{Gamma}(\alpha, \beta) \\ \mu|\tau &\sim \text{Normal}\left(\xi, \frac{1}{\phi\tau}\right) \\ w_n|\mu,\tau &\sim \text{Normal}\left(\mu, \frac{1}{\tau}\right), \qquad\qquad n = 1, \dots, N \end{aligned}$$

where α, β, ξ, ϕ are hyper-parameters of known values as before.

To generate posterior samples $r^{(j)} = \{\mu^{(j)}, \tau^{(j)}\}$, this time using a Metropolis-Hastings scheme, we may consider proposals of the form

$$\mathbb{Q}_{\mu^{\text{old}},\tau^{\text{old}}} = \text{Normal}_2\left(\begin{bmatrix}\mu^{\text{old}} \\ \tau^{\text{old}}\end{bmatrix}, \begin{bmatrix}\lambda_\mu^2, 0 \\ 0, \lambda_\tau^2\end{bmatrix}\right)$$

that is, the proposals $\mu^{\text{prop}}, \tau^{\text{prop}}$ are sampled jointly from a bivariate normal with mean on the previous sample and some pre-set variances λ_μ^2 and λ_τ^2. With this choice, the acceptance log-ratio becomes

$$L_{\mu^{\text{old}},\tau^{\text{old}}}(\mu^{\text{prop}}, \tau^{\text{prop}}) = \begin{cases} \frac{1}{2}\sum_{n=1}^{N}\left(\tau^{\text{old}}(\mu^{\text{old}} - w_n)^2 - \tau^{\text{prop}}(\mu^{\text{prop}} - w_n)^2\right) \\ +\frac{N+2\alpha-1}{2}\log\frac{\tau^{\text{prop}}}{\tau^{\text{old}}} \\ +\frac{\phi}{2}\left(\tau^{\text{old}}(\mu^{\text{old}} - \xi)^2 - \tau^{\text{prop}}(\mu^{\text{prop}} - \xi)^2\right) \\ +\beta\left(\tau^{\text{old}} - \tau^{\text{prop}}\right), & \tau^{\text{prop}} > 0 \\ \\ -\infty, & \tau^{\text{prop}} \leq 0 \end{cases}.$$

As an illustrative example consider a total of $N = 10$ observations w_n, generated through Normal$(5, 1)$. Such observations are similar to example 5.1 and, for completeness, are also shown in fig. 5.3A. As before, we also use the same hyper-parameters: $\alpha = 2, \beta = 1, \xi = 2, \phi = 1$. A total of $J = 500$ samples are generated with $\lambda_\mu^2 = \lambda_\tau^2 = 1$, and shown in fig. 5.3BC. As can be seen, although the sampler starts at a point $\mu^{(0)} = 1, \tau^{(0)} = 1$ of low posterior probability, the chain quickly moves towards and eventually remains near the posterior mode located around $\mu = 4.5, \tau = 1$.

The Metropolis-Hastings sampler is a powerful tool in Bayesian inference as it can be used to sample from *any* posterior irrespective of whether the prior and likelihood are conjugate. Nevertheless, its practical implementation in a complex setting is often challenging as the proposal $Q_{r^{\text{old}}}(r^{\text{prop}})$ used dictates the algorithm's performance. To wit, it is always possible to devise theoretically valid proposals, but a judicious choice and extensive calibration are typically critical requirements for the algorithm to appropriately sample from the posterior in a reasonable computational time. Unfortunately, with most naive choices, the number of MCMC samples, J, required to adequately characterize a target may be exuberantly large and, even for moderate scale applications, may involve infeasible computational cost.

Example 5.4: Choice of proposals in MCMC

To illustrate just how important the proposal is in allowing the sampler to cover the entire support of the target within few iterations, we consider a bivariate target $\pi(x, y)$ with a mixture form

$$\mathbb{\pi} = 0.3\,\text{Normal}_2\left(\begin{bmatrix}-1 \\ 0\end{bmatrix}, \begin{bmatrix}0.25^2 & 0 \\ 0 & 0.25^2\end{bmatrix}\right)$$

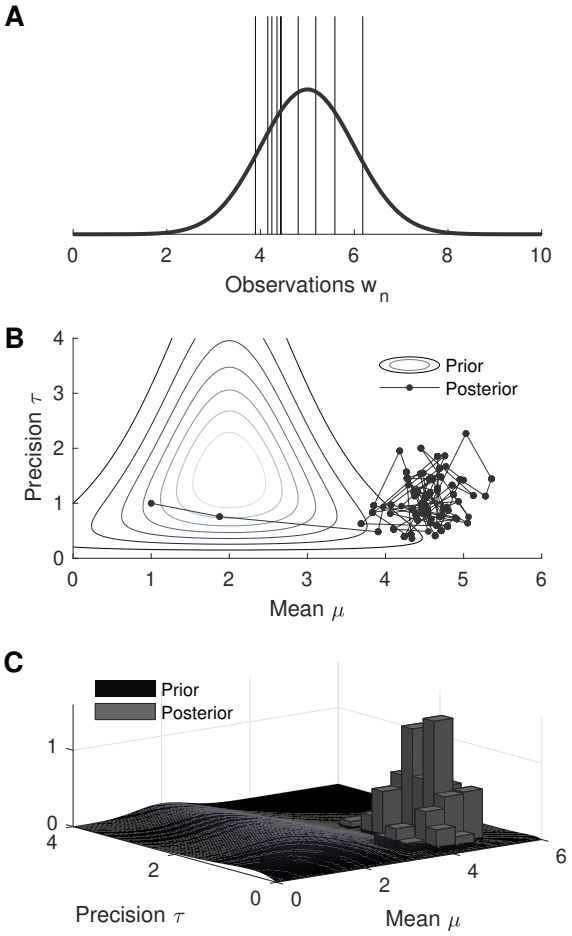


Figure 5.3: Illustration of example 5.3. Panel A: observations $w_{1:N}$ (vertical lines) and generating distribution (solid line). Panel B: prior probability distribution $p(\mu, \tau)$ (contours) and MCMC samples $(\mu^{(j)}, \tau^{(j)})$ from the posterior $p(\mu, \tau | w_{1:N})$ (dots). Panel C: comparison of prior $p(\mu, \tau)$ (surface) and posterior $p(\mu, \tau | w)$ (histogram).

$$+ 0.3\,\text{Normal}_2\left(\begin{bmatrix}+1\\0\end{bmatrix}, \begin{bmatrix}0.25^2 & 0\\0 & 0.25^2\end{bmatrix}\right)$$
$$+ 0.4\,\text{Normal}_2\left(\begin{bmatrix}0\\+1\end{bmatrix}, \begin{bmatrix}0.25^2 & 0\\0 & 0.25^2\end{bmatrix}\right)$$

consisting of three well separated modes. Although there are much more efficient methods to simulate from such a target, here we consider a naive Metropolis-Hastings scheme with proposal

$$\mathbb{Q}_{x^{\text{old}},y^{\text{old}}} = \text{Normal}_2\left(\begin{bmatrix}x^{\text{old}}\\y^{\text{old}}\end{bmatrix}, \begin{bmatrix}0.3^2 & 0\\0 & 0.3^2\end{bmatrix}\right).$$

Figure 5.4 illustrates MCMC approximations of the target produced by two different chains: a shorter chain containing $J = 5 \times 10^2$ samples and a longer one containing $J = 5 \times 10^6$ samples. For this choice of proposal, only the latter chain samples all three modes.

For this illustrative example, each iteration of the sampler comes at low computation cost; the most costly operations being the two Normal draws. As such, a large number of samples, such as $\approx 10^6$ in the second chain, can be obtained in reasonable time. However, in more complex applications, where each iteration may involve many costly operations, this number of samples may remain purely aspirational.

Metropolis sampler

The Metropolis sampler is a special version of the Metropolis-Hastings sampler that we already saw in the previous section and its implementation is identical to algorithm 5.1. In this version of the sampler, the proposal $Q_{r^{\text{old}}}(r^{\text{prop}})$ is symmetric with respect to r^{old} and r^{prop}, *i.e.* $Q_{r^{\text{old}}}(r^{\text{prop}}) = Q_{r^{\text{prop}}}(r^{\text{old}})$ for any feasible r^{old} and r^{prop}. Due to this symmetry, the acceptance ratio simplifies to

$$A_{r^{\text{old}}}(r^{\text{prop}}) = \frac{\bar{\pi}(r^{\text{prop}})}{\bar{\pi}(r^{\text{old}})}, \tag{5.11}$$

since the last factor in eq. (5.9) now becomes unity. Similarly, when the target $\bar{\pi}(r)$ is the posterior of a Bayesian model, eq. (5.10) reduces to

$$A_{r^{\text{old}}}(r^{\text{prop}}) = \underbrace{\frac{p(w|r^{\text{prop}})}{p(w|r^{\text{old}})}}_{\text{likelihood}} \underbrace{\frac{p(r^{\text{prop}})}{p(r^{\text{old}})}}_{\text{prior}}. \tag{5.12}$$

Example 5.5: MCMC under a Cauchy prior

Consider scalar observations $w_{1:N}$ normally distributed around some unknown mean μ. For convenience, also assume that w_n and μ are scaled such that the variance is 1. In this example, the goal is to estimate μ, but instead of making the common prior choice, we place a Cauchy distribution which is much wider. The entire model is

$$\begin{aligned}\mu &\sim \text{Cauchy}(0,1)\\ w_n|\mu &\sim \text{Normal}(\mu,1), \qquad\qquad n = 1,\ldots,N.\end{aligned}$$

Under this model, the posterior is

$$\pi(\mu) = p(\mu|w_{1:N}) \propto \left[\prod_n p(w_n|\mu)\right] p(\mu) \propto \frac{\exp\left(-\frac{1}{2}\sum_n (w_n-\mu)^2\right)}{1+\mu^2}.$$

Although we can easily visualize this posterior due to the analytic form above, it is still hard to quantify point estimates from our posterior since we cannot analytically compute expectations with such a density. Instead, we

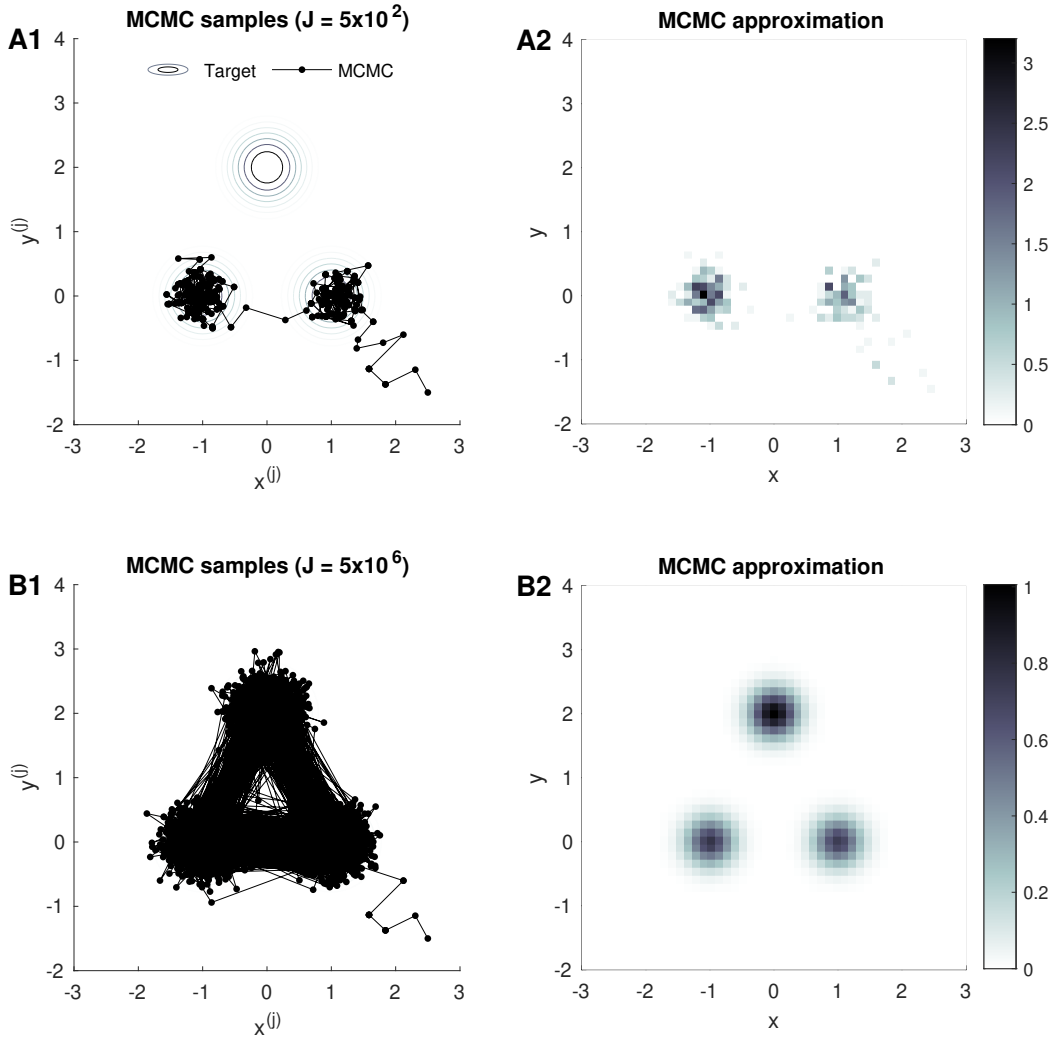


Figure 5.4: Illustration of example 5.4. Two Metropolis-Hastings MCMC chains are computed for the evaluation of the same target. For the chain in panel A1 and the corresponding histogram, only $J = 5 \times 10^2$ samplers are used. By contrast, the chain in panel B2 and the corresponding histogram, contains $J = 5 \times 10^6$ samples. As can be seen, only the latter chain fully samples the target.

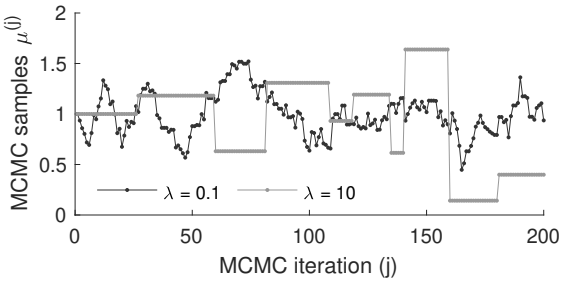


Figure 5.5: Two Metropolis samplers on the same target. In the long run both chains cover the entire support of the target; however, the total number of samples required differs substantially.

may use a Metropolis sampler with proposals

$$\mathbb{Q}_{\mu^{\text{old}}} = \text{Normal}(\mu^{\text{old}}, \lambda^2)$$

to generate MCMC samples $\mu^{(j)}$ that are distributed according to $\pi(\mu)$. Once a significant number J of $\mu^{(j)}$ is obtained, using eq. (5.2), we can compute any expectation, for example

$$\langle \mu \rangle_\pi = \int_{-\infty}^{+\infty} d\mu\, \mu \pi(\mu) \approx \frac{1}{J} \sum_{j=1}^{J} \mu^{(j)}.$$

Figure 5.5 illustrates two MCMC chains obtained with different values of λ targeting the same posterior $\pi(\mu)$. Both chains start at $\mu^{(0)} = 1$ and continued for a total of $J = 200$ samples. As can be seen a different total number of samples J is needed so each chain adequately covers the target's support.

Additive random walk sampler

A straightforward approach to obtaining Metropolis proposals $\mathbb{Q}_{r^{\text{old}}}$ is to consider local explorations around r^{old}. We have already used this idea in examples 5.2 to 5.4 and here we formalize our description.

A natural way of obtain proposals capable of local exploration is through (additive) *random walks*

$$r^{\text{prop}} = r^{\text{old}} + \epsilon^{\text{prop}},$$

where ϵ^{prop} is a random perturbation sampled from a density $G(\epsilon^{\text{prop}})$ that is independent of r^{old}. For convenience, *univariate* perturbations ϵ^{prop} can be sampled from Normal, StudentT, or Uniform distributions. Maintaining symmetry requires $G(-\epsilon) = G(+\epsilon)$, and for practical reasons the perturbations need to be scaled. Precisely, in a unified setting, convenient choices may be

$$\frac{\epsilon^{\text{prop}}}{\lambda} \sim \text{Normal}(0, 1), \qquad \frac{\epsilon^{\text{prop}}}{\lambda} \sim \text{StudentT}(\nu), \qquad \frac{\epsilon^{\text{prop}}}{\lambda} \sim \text{Uniform}_{[-1,+1]},$$

where $\lambda > 0$ is a scaling constant that characterizes the "spread" of the perturbation. In other words, λ controls the "volume" around r^{old} to be explored each time. These perturbations are demonstrated in fig. 5.6. Of course, when a *multivariate* proposal r^{prop} is required, *e.g.* as in example 5.3, the univariate perturbations shown above can be readily generalized.

Note 5.8: What is *not* a Metropolis sampler?

So far we have mostly seen instances of Metropolis samplers. In most examples, the proposals $Q_{r^{\text{old}}}(r^{\text{prop}})$ have been of the special forms mentioned above. Naively, one might conclude that every distribution in these families

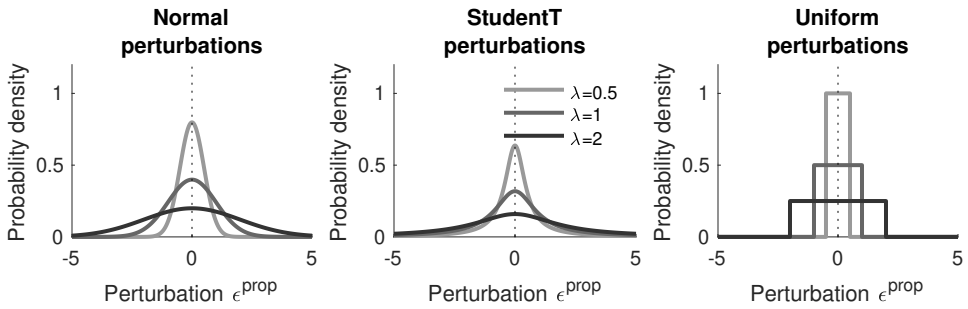


Figure 5.6: Probability densities of common univariate Metropolis random walk perturbations. The values of the scaling constant λ determines the spread of the perturbation. StudentT distributions shown in the middle panel use $\nu = 2$ degrees of freedom.

might be suitable for a Metropolis sampler. To demonstrate that this is *not* necessarily true, we consider proposals of the form

$$\mathbb{Q}_{r^{\text{old}}} = \text{Normal}\left(r^{\text{old}}, (r^{\text{old}})^2\right).$$

Here, the spread of the proposal is not constant, *i.e.* both the proposal's mean and variance depend on r^{old}. As a result, the proposal cannot be symmetric because, in general,

$$\text{Normal}\left(r^{\text{prop}}; r^{\text{old}}, (r^{\text{old}})^2\right) \neq \text{Normal}\left(r^{\text{old}}; r^{\text{prop}}, (r^{\text{prop}})^2\right).$$

So, the ratio of the proposals $Q_{r^{\text{prop}}}(r^{\text{old}})/Q_{r^{\text{old}}}(r^{\text{prop}})$ does not drop from $A_{r^{\text{old}}}(r^{\text{old}})$ in eq. (5.9) as needs to happen in a Metropolis sampler.

In most applications, it is more convenient to preselect the perturbation's density $G(\epsilon)$ and subsequently calibrate λ, which, as we show in example 5.6, has an important effect on the resulting MCMC chain. Irrespective of calibration, however, random walk samplers tend to perform very poorly on targets that depend on more than just a handful of variables.

Example 5.6: Scaling a MCMC random walk

We consider a univariate target $\pi(r)$ consisting of a mixture of two Normal densities

$$\mathbb{\pi} = 0.7\,\text{Normal}(+1, 0.04) + 0.3\,\text{Normal}(-1, 0.04).$$

Although there exist more efficient choices to sample from such a target, here we consider a Metropolis random walk with uniform perturbations $\mathbb{G} = \text{Uniform}_{[-1,+1]}$. Under a scaling λ, the corresponding proposals are given by

$$\mathbb{Q}_{r^{\text{old}}} = \text{Uniform}_{[r^{\text{old}}-\lambda, r^{\text{old}}+\lambda]}.$$

Figure 5.7 shows three MCMC chains obtained with $\lambda = 0.5, 5, 50$. To facilitate a comparison, we start all chains at $r^{(0)} = 1$ and continue for up to $J = 500$ samples.

As can be seen, the chains with low λ move very slowly through the support of $\pi(r)$; while with higher λ, as expected, show higher motility. Consequently, within the first 500 samples computed, only the last two chains explored a region large enough to cover the entire support of $\pi(r)$. This is a common characteristic of random walks and it is particularly pronounced when the target consists of more than one mode separated by regions of low probability. In such cases, extremely large numbers of samples J may be required until the chain fully explores the entire posterior.

We may naively select larger values of λ and try to avoid a sampler that might be easily trapped. However, as can be seen in the last case, larger λ values tend to suffer high rejection rates, which can be deduced by the large intervals over which a chain remains constant. There are two drawbacks associated with high rejections: (i) the

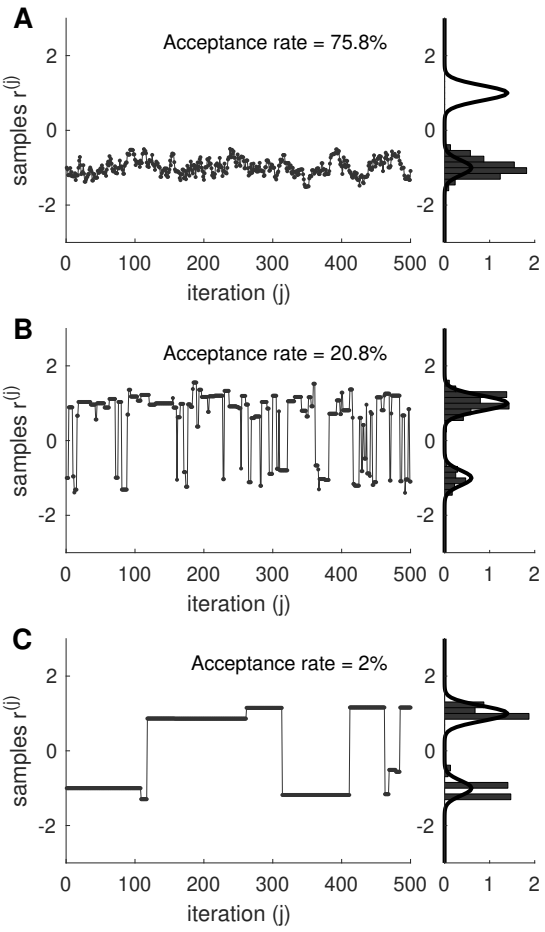


Figure 5.7: Demonstration of the effects of a random walk's scaling λ on the resulting MCMC chain. Low λ values are associated with high acceptance rates and slowly evolving chains that may be easily trapped near a target's modes. Intermediate λ values are associated with fair acceptance rates and quickly evolving chains that sporadically jump among modes. High λ values are associated with low acceptance rates and chains with large, but rare steps, that easily jumps among modes, however, without revealing much of the target's finer details.

chain stays at the same values often and so a large number of samples are required in order to reveal a target's fine details; (ii) as the computational cost of producing a rejected proposal is the same as an accepted proposal, large λ leads to unreasonably costly samplers.

Therefore, it is not surprising that intermediate λ's are most efficient as these sufficiently sample the entire target $\pi(r)$ while maintaining reasonable rejection levels.

5.2.2 Gibbs family of samplers

The samplers in the Gibbs family are useful only in cases where the targeted random variable is *multivariate* and partitioned into blocks of component variables. For this reason, in this section we adopt vector notation, and denote the random variable of interest with $\boldsymbol{R} = R_{1:M}$, where R_m are the individual components. Of course, it is not required that each component R_m be univariate. In fact, it may even be multivariate itself. Accordingly, $\pi(\boldsymbol{R}) = \pi(R_{1:M}) = \pi(R_1, \ldots, R_M)$ all denote our targeted density and similarly for their unnormalized counterparts.

The Gibbs sampler is the preferred choice when sampling a multivariate target especially when there is a

natural partitioning of the various components in groups R_m. The basic idea is that, to sample $\boldsymbol{R}$, it is sufficient to sample each R_m separately. Before we describe how and why the Gibbs sampler works, we introduce some useful notation.

For simplicity, similarly to other chapters, we use subscripts to group our variable's components

$$r_{1:m} = (r_1, \ldots, r_m), \qquad r_{m:M} = (r_m, \ldots, r_M), \qquad r_{-m} = (r_1, \ldots, r_{m-1}, r_{m+1}, \ldots, r_M).$$

Specifically in r_{-m}, we use *negative* subscripts to group all components *but* r_m. Additionally, we use $\Pi^m_{r_{-m}}$ to denote the distribution of $R_m|r_{-m}$ and $\pi^m_{r_{-m}}(r_m)$ to denote its density; that is

$$R_m|r_{-m} \sim \Pi^m_{r_{-m}}, \qquad \pi^m_{r_{-m}}(r_m) = p(r_m|r_{-m}).$$

These are termed *full conditionals* and our targeted variable $\boldsymbol{R}$ entails M of them; one for each m. For clarity, we will use a superscript m to emphasize explicitly that they refer to the m^{th} component of $\boldsymbol{R}$ and a subscript r_{-m} to emphasize that these condition on every *other* component's value.

Gibbs sampler

Given a multivariate target $\bar{\pi}(\boldsymbol{r})$, to begin a Gibbs sampler, a choice for the initial sample $\boldsymbol{r}^{(0)}$ needs to be made. Similar to the Metropolis-Hastings sampler, this can be achieved either by assigning a fixed value or sampling a value from some probability distribution. As before, the only requirement is that a feasible value be selected, $\bar{\pi}(\boldsymbol{r}^{(0)}) > 0$.

To advance the chain from $\boldsymbol{r}^{\text{old}} = r^{\text{old}}_{1:M}$ to $\boldsymbol{r}^{\text{new}} = r^{\text{new}}_{1:M}$, the Gibbs sampler uses M stages during which each component R_m is generated separately. In particular, the scheme proceeds by sampling each r_m conditioning on the most recent value of r_{-m} as follows

Stage 1	$R^{\text{new}}_1 \vert r^{\text{old}}_{2:M} \sim \Pi^1_{r^{\text{old}}_{2:M}}$
Stage 2	$R^{\text{new}}_2 \vert r^{\text{new}}_1, r^{\text{old}}_{3:M} \sim \Pi^2_{r^{\text{new}}_1, r^{\text{old}}_{3:M}}$
$\ldots$	$\ldots$
Stage $M-1$	$R^{\text{new}}_{M-1} \vert r^{\text{prop}}_{1:M-2}, r^{\text{old}}_M \sim \Pi^{M-1}_{r^{\text{new}}_{1:M-1}, r^{\text{old}}_M}$
Stage M	$R^{\text{new}}_M \vert r^{\text{prop}}_{1:M-1} \sim \Pi^M_{r^{\text{new}}_{1:M-1}}$

That is, every component r^{new}_m is obtained by sampling from the corresponding full conditional distribution conditioned on the most recently available r_{-m}. This consists of a combination of already updated components $r^{\text{new}}_{1:m-1}$ and those that have not been updated yet $r^{\text{old}}_{m+1:M}$. For this reason, informally we might refer to each stage as an *update*.

In example 5.7, we provide a demonstration of the algorithm's implementation and we explicitly illustrate each of its stages. In algorithm 5.2, we summarize a computational implementation of the entire sampling scheme.

Example 5.7: Gibbs sampling of a tri-variate target

We consider a special case where $\boldsymbol{R} = (R_1, R_2, R_3)$ consists of three components. In this case, the target and full conditionals are

$$\begin{aligned}
\pi(r_1, r_2, r_3) &= p(r_1, r_2, r_3) \\
\pi^1_{r_2, r_3}(r_1) &= p(r_1|r_2, r_3) \\
\pi^2_{r_1, r_3}(r_2) &= p(r_2|r_1, r_3) \\
\pi^3_{r_1, r_2}(r_3) &= p(r_3|r_1, r_2).
\end{aligned}$$

The Gibbs sampler uses an already computed sample $\boldsymbol{r}^{\text{old}} = (r^{\text{old}}_1, r^{\text{old}}_2, r^{\text{old}}_3)$ to obtain $\boldsymbol{r}^{\text{new}} = (r^{\text{new}}_1, r^{\text{new}}_2, r^{\text{new}}_3)$

by first generating

$$r_1^{\text{new}} \sim \mathbb{\Pi}^1_{r_2^{\text{old}}, r_3^{\text{old}}},$$

subsequently generating

$$r_2^{\text{new}} \sim \mathbb{\Pi}^2_{r_1^{\text{new}}, r_3^{\text{old}}},$$

and finally generating

$$r_3^{\text{new}} \sim \mathbb{\Pi}^3_{r_1^{\text{new}}, r_2^{\text{new}}}.$$

These three updates require sampling from the full conditionals $\mathbb{\Pi}^1_{r_2,r_3}, \mathbb{\Pi}^2_{r_1,r_3}, \mathbb{\Pi}^3_{r_2,r_3}$, respectively, which need to be specified in advance.

Algorithm 5.2: Gibbs sampler (fixed sweep)

Given the full conditionals $\pi^m_{r_{-m}}(r_m)$ of a target and a feasible initial sample $\boldsymbol{r}^{(0)}$, the Gibbs sampler proceeds as follows:
For each j from 1 to J repeat:

- For each m from 1 up to M repeat:
  - Sample $r_m^{(j)} \sim \mathbb{\Pi}^m_{r_{-m}}$ conditioning on the most recent r_{-m}.

In a Gibbs scheme, as described above, the updates of the components r_m do *not* need to take place in the given order (fixed sweep) as in algorithm 5.2. In fact, the sampler remains valid even if the order of the updates is chosen at random (random sweep) as we summarize in algorithm 5.3. In both version, updating of each R_m is achieved by $\mathbb{\Pi}^m_{-r_m}$ conditioned on the most recent r_{-m}. The difference between the two versions is only on the order at which the sampler sweeps through $R_1, \ldots, R_M$.

Algorithm 5.3: Gibbs sampler (random sweep)

Given the full conditionals $\pi^m_{r_{-m}}(r_m)$ of a target and a feasible initial sample $\boldsymbol{r}^{(0)}$, the Gibbs sampler proceeds as follows:
For each j from 1 to J repeat:

- Generate a permutation $\sigma_{1:M}$ of $1, \ldots, M$
- For each m from 1 up to M repeat:
  - Sample $r_{\sigma_m}^{(j)} \sim \mathbb{\Pi}^{\sigma_m}_{r_{-\sigma_m}}$ conditioning on the most recent $r_{-\sigma_m}$.

A Gibbs scheme, either in the fixed or random sweep version, relies on the generation of samples from the full conditionals $\mathbb{\Pi}^m_{r_{-m}}$. For this reason, it is crucial that the full conditionals be simulated directly. The choice of the partitioning of $\boldsymbol{R}$ into groups $R_{1:M}$ has a significant impact on the sampler as it, essentially, determines whether the resulting conditionals $\mathbb{\Pi}^m_{r_{-m}}$ may be simulated. In section 5.2.2, we will see an alternative scheme, however less efficient, that can be used when some of the conditionals cannot be simulated directly.

Why the sampler works?*

Given the full conditionals $\pi^m_{r_{-m}}(r_m)$ of a target $\pi(\boldsymbol{r})$, the Gibbs sampler ensures that the MCMC chain generated visits each feasible $\boldsymbol{r}$ with a frequency proportional to $\pi(\boldsymbol{r})$. Similar to the Metropolis-Hastings sampler we saw earlier, this entails a sophisticated bookkeeping mechanism that balances flow in and out of each $\boldsymbol{r}$ which we will

*This is an advanced topic and could be skipped on a first reading.

illustrate in detail. As before, we will investigate thoroughly this mechanism. Specifically, we will spell out the transition rules and show that they are in balance with our target.

In either of its versions, the Gibbs sampler advances from a sample $\boldsymbol{r}^{(j)}$ to the next one $\boldsymbol{r}^{(j+1)}$ in the MCMC chain through the successive realization of M intermediate samples

$$\boldsymbol{R}^{(j)} = \boldsymbol{R}^{(j,0)} \to \underbrace{\boldsymbol{R}^{(j,1)}}_{\substack{1^{\text{st}}\\ \text{update}}} \to \underbrace{\boldsymbol{R}^{(j,2)}}_{\substack{2^{\text{nd}}\\ \text{update}}} \to \cdots \to \underbrace{\boldsymbol{R}^{(j,M)}}_{\substack{\text{M}^{\text{th}}\\ \text{update}}} = \boldsymbol{R}^{(j+1)}.$$

The precise order of the components sampled in each update depends on the version of the sampler chosen. Nevertheless, both versions entail intermediate transition rules that update a single component each. For clarity, from now on, we will denote the rule that updates the component r_m with $T^m_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}^{\text{new}})$. As $T^m_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}^{\text{new}})$ leaves r_{-m} unchanged, it attains the specific form

$$T^m_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}^{\text{new}}) = \pi^m_{r^{\text{old}}_{-m}}(r^{\text{new}}_m)\delta_{r^{\text{old}}_{-m}}(r^{\text{new}}_{-m}).$$

Transition rules First, we focus on a single Gibbs update. This is equivalent to a Metropolis-Hastings scheme with proposals

$$Q_{r^{\text{old}}_m, r^{\text{old}}_{-m}}(r^{\text{prop}}_m, r^{\text{prop}}_{-m}) = \pi^m_{r^{\text{old}}_{-m}}(r^{\text{prop}}_m)\delta_{r^{\text{old}}_{-m}}(r^{\text{prop}}_{-m}).$$

To verify that, indeed, the two schemes are equivalent, we consider the acceptance ratio

$$\begin{aligned} A^m_{r^{\text{old}}_m, r^{\text{old}}_{-m}}(r^{\text{prop}}_m, r^{\text{prop}}_{-m}) &= \frac{\pi(r^{\text{prop}}_m, r^{\text{prop}}_{-m})}{\pi(r^{\text{old}}_m, r^{\text{old}}_{-m})} \frac{Q_{r^{\text{prop}}_m, r^{\text{prop}}_{-m}}(r^{\text{old}}_m, r^{\text{old}}_{-m})}{Q_{r^{\text{old}}_m, r^{\text{old}}_{-m}}(r^{\text{prop}}_m, r^{\text{prop}}_{-m})} \\ &= \underbrace{\frac{\pi^m_{r^{\text{prop}}_{-m}}(r^{\text{prop}}_m)}{\pi^m_{r^{\text{old}}_{-m}}(r^{\text{old}}_m)} \frac{\pi^{-m}(r^{\text{prop}}_{-m})}{\pi^{-m}(r^{\text{old}}_{-m})}}_{\text{target}} \underbrace{\frac{\pi^m_{r^{\text{prop}}_{-m}}(r^{\text{old}}_m)}{\pi^m_{r^{\text{old}}_{-m}}(r^{\text{prop}}_m)} \frac{\delta_{r^{\text{prop}}_{-m}}(r^{\text{old}}_{-m})}{\delta_{r^{\text{old}}_{-m}}(r^{\text{prop}}_{-m})}}_{\text{proposal}}. \end{aligned}$$

Here, we use $\pi^{-m}(r_{-m})$ to denote the probability density of R_{-m} with R_m marginalized. As the proposal does not change r_{-m}, we have $r^{\text{prop}}_{-m} = r^{\text{old}}_{-m}$. Therefore, the acceptance ratio simplifies to $A^m_{r^{\text{old}}_m, r^{\text{old}}_{-m}}(r^{\text{prop}}_m, r^{\text{prop}}_{-m}) = 1$. Consequently, $r^{\text{new}}_m = r^{\text{prop}}_m$ and the acceptance test need not be performed since the proposal is certainly accepted.

We now consider the application of two successive Gibbs updates. For instance, updates m and $m+1$ in the fixed sweep version. In this case, the first update implements a transition $\boldsymbol{r}^{\text{old}} \to \boldsymbol{r}^{\text{temp}}$ to some intermediate $\boldsymbol{r}^{\text{temp}}$; while, the second update implements the transition $\boldsymbol{r}^{\text{temp}} \to \boldsymbol{r}^{\text{new}}$. To derive what the combine transition rule $T^{m,m+1}_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}^{\text{new}})$ looks like, we need to consider all possibilities for the intermediate sample. In other words we need to marginalize over the intermediate variable $\boldsymbol{R}^{\text{temp}}$. Formally,

$$\begin{aligned} T^{m,m+1}_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}^{\text{new}}) = p\left(\boldsymbol{r}^{\text{new}}|\boldsymbol{r}^{\text{old}}\right) &= \int_{\boldsymbol{r}^{\text{temp}}} d\boldsymbol{r}^{\text{temp}}\, p\left(\boldsymbol{r}^{\text{new}}, \boldsymbol{r}^{\text{temp}}|\boldsymbol{r}^{\text{old}}\right) \\ &= \int_{\boldsymbol{r}^{\text{temp}}} d\boldsymbol{r}^{\text{temp}}\, p\left(\boldsymbol{r}^{\text{temp}}|\boldsymbol{r}^{\text{old}}\right) p\left(\boldsymbol{r}^{\text{new}}|\boldsymbol{r}^{\text{temp}}\right) = \int_{\boldsymbol{r}^{\text{temp}}} d\boldsymbol{r}^{\text{temp}}\, T^m_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}^{\text{temp}}) T^{m+1}_{\boldsymbol{r}^{\text{temp}}}(\boldsymbol{r}^{\text{new}}). \end{aligned}$$

With similar completions, we can derive the transition rules $T_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}^{\text{new}})$ of the entire Gibbs scheme. These, of course, entail all M updates and the precise order they are applied depends on the particular version of the sampler.

Balance conditions Because each Gibbs update results from a Metropolis-Hastings scheme, the associated rule $T^m_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}^{\text{new}})$ is already in full balance with the target $\pi(\boldsymbol{r})$. This means that

$$\pi(\boldsymbol{r}) = \int_{\boldsymbol{r}^{\text{old}}} d\boldsymbol{r}^{\text{old}}\, T^m_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r})\pi(\boldsymbol{r}^{\text{old}}),$$

holds for all m, which is an important result we need to derive the balance condition for the entire Gibbs scheme. To this end, we first consider the application of only two successive Gibbs updates, for instance m and $m+1$ as above. In this case, we have

$$\begin{aligned}\int_{\boldsymbol{r}^{\text{old}}} d\boldsymbol{r}^{\text{old}}\, T^{m,m+1}_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}^{\text{new}})\pi(\boldsymbol{r}^{\text{old}}) &= \int_{\boldsymbol{r}^{\text{old}}} d\boldsymbol{r}^{\text{old}} \left(\int_{\boldsymbol{r}} d\boldsymbol{r}\, T^{m}_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}) T^{m+1}_{\boldsymbol{r}}(\boldsymbol{r}^{\text{new}})\right) \pi(\boldsymbol{r}^{\text{old}}) \\ &= \int_{\boldsymbol{r}} d\boldsymbol{r} \left(\int_{\boldsymbol{r}^{\text{old}}} d\boldsymbol{r}^{\text{old}}\, T^{m}_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r})\pi(\boldsymbol{r}^{\text{old}})\right) T^{m+1}_{\boldsymbol{r}}(\boldsymbol{r}^{\text{new}}) \\ &= \int_{\boldsymbol{r}} d\boldsymbol{r}\, \pi(\boldsymbol{r}) T^{m+1}_{\boldsymbol{r}}(\boldsymbol{r}^{\text{new}}) \\ &= \pi(\boldsymbol{r}^{\text{new}}),\end{aligned}$$

which shows that $T^{m,m+1}_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}^{\text{new}})$ is also in full balance with the target $\pi(\boldsymbol{r})$. We can similarly show that the transition rules $T_{\boldsymbol{r}^{\text{old}}}(\boldsymbol{r}^{\text{new}})$ of the entire Gibbs scheme are also in full balance with the target $\pi(\boldsymbol{r})$.

Of course, the full balance condition fulfilled by the Gibbs sampler is a weaker condition than detailed balance fulfilled by the Metropolis-Hastings sampler. Nevertheless, in view of note 5.3, it is sufficient to ensure that the resulting MCMC chain is valid. Exercise 5.8 explores some further balance properties of the Gibbs sampler.

Example 5.8: When does the Gibbs sampler fail?

Consider sampling a bivariate random variable (R_1, R_2) from a target $\pi(r_1, r_2)$ that equals to $1/2$ whenever $-1 < r_1, r_2 < 0$ or $0 < r_1, r_2 < +1$. Essentially this is a Uniform target over two disjoint rectangular regions; see fig. 5.8. Because this is a simple target, we can easily obtain the full conditionals

$$\pi^1_{r_2}(r_1) = \begin{cases} \text{Uniform}_{[-1,0]}(r_1), & r_2 < 0 \\ \text{Uniform}_{[0,+1]}(r_1), & 0 < r_2 \end{cases}$$

$$\pi^2_{r_1}(r_2) = \begin{cases} \text{Uniform}_{[-1,0]}(r_2), & r_1 < 0 \\ \text{Uniform}_{[0,+1]}(r_2), & 0 < r_1 \end{cases}$$

For concreteness, consider a Gibbs sampler starting at some $r_1^{(0)} < 0$ and $r_2^{(0)} < 0$. This sample lies in the left region. As can be seen, following the first iteration

- sample $r_1^{(1)}|r_2^{(0)} \sim \text{Uniform}_{[-1,0]}(r_1)$
- sample $r_2^{(1)}|r_1^{(1)} \sim \text{Uniform}_{[-1,0]}(r_1)$

the sampler remains in the same region. The same happens following the second iteration

- sample $r_1^{(2)}|r_2^{(1)} \sim \text{Uniform}_{[-1,0]}(r_1)$
- sample $r_2^{(2)}|r_1^{(2)} \sim \text{Uniform}_{[-1,0]}(r_1)$

and every subsequent one. Accordingly, no matter how may iterations we perform, the sampler is unable to cross into the other region. The same happens also when the sampler is initialized with $r_1^{(0)} > 0$ and $r_2^{(0)} > 0$ in which case it remains trapped in the right region.

Recalling note 5.3, we see that clearly this sampler meets the *feasibility* condition. Additionally, as we show above, it also meets the *invariance* condition. Nevertheless, it does *not* meet the *irreducibility* condition. In particular, because each transition $\boldsymbol{r}^{\text{old}} \to \boldsymbol{r}^{\text{new}}$ is broken down to two separate transitions, neither of which can cross into the other region, the resulting sampler is non-ergodic.

Generally Gibbs samplers encounter problems whenever the support of the target is disconnected. The same challenge persists even when the target's support is "effectively" disconnected as, for instance, when it contains ridges of very low probability. In exercise 5.10, we show that ergodicity can be recovered by a reparametrization of the target. Such an approach suggests that reparametrizarion of the target offers a general remedy; however, in complex targets choosing an appropriate parametrization is a demanding task.

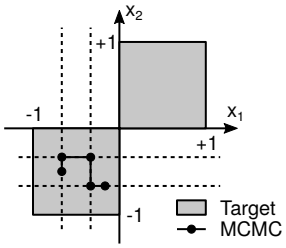


Figure 5.8: An example illustrating that Gibbs sampling may fail to ergodically explore the support of the target.

Sampling of posterior targets

Just as with Metropolis-Hastings, the Gibbs sampler holds equally well whenever our target is a Bayesian posterior $\pi(\boldsymbol{r}) = p(\boldsymbol{r}|w)$. The full conditionals in this case read

$$\pi^m_{r_{-m}}(r_m) = p(r_m|r_{-m}, w).$$

We emphasize that, unlike a general target in which observations w are irrelevant, the full conditionals of a Bayesian posterior are conditioned also on the observations w. As we demonstrate in subsequent examples, direct sampling from such conditionals $\pi^m_{r_{-m}}(r_m)$ is particularly easy in Bayesian models developed in terms of conditionally conjugate distributions.

Example 5.9: MC for the conditionally conjugate Normal-Gamma model

In this example we consider a variant of the model of example 5.1. As before, suppose we have scalar observations w_n, for $n = 1, \ldots, N$, that are normally distributed and, again, our task is to estimate the center and the spread of the underlying distribution. For this, consider a similar parametrization of the Normal distribution by mean μ and precision τ. However, unlike before where the prior on μ and τ was jointly specified (note the conditional on eq. (5.4)), here we apply independent priors on the two parameters. In particular, we place a Normal prior on μ and a gamma prior on τ, so the entire model becomes

$$\tau \sim \text{Gamma}(\alpha, \beta) \tag{5.13}$$

$$\mu \sim \text{Normal}(\xi, \psi) \tag{5.14}$$

$$w_n|\mu, \tau \sim \text{Normal}\left(\mu, \frac{1}{\tau}\right), \qquad n = 1, \ldots, N \tag{5.15}$$

where α, β, ξ, ψ are hyper-parameters of known values for simplicity.

With this setup, the model is described by the random variable $\boldsymbol{r} = (\mu, \tau)$, and the associated posterior is $\pi(\boldsymbol{r}) = p(\mu, \tau|w_{1:N})$. Despite its similarity to example 5.1, the new model is more cumbersome. Namely, a minor change in the description of the prior (eq. (5.4) vs. eq. (5.13)) has a major impact for practical applications, as now the new model is no longer fully conjugate. Therefore computational methods are required to characterize the induced posterior. By contrast, the model is conditionally conjugate which facilitates Gibbs sampling. For instance, a Gibbs sampler requires two stages: one to update μ and one to update τ. Suppose that $\boldsymbol{r}^{(j)} = (\mu^{(j)}, \tau^{(j)})$ has been computed, so the updates are

- sample $\mu^{(j+1)}$ from $\mu|\tau^{(j)}, w_{1:N}$
- sample $\tau^{(j+1)}$ from $\tau|\mu^{(j+1)}, w_{1:N}$

Once both parameters have been updated, the new sample is $r^{(j+1)} = (\mu^{(j+1)}, \tau^{(j+1)})$.

The required conditionals for each update may be worked out analytically

$$p\left(\mu|\tau, w_{1:N}\right) \propto p\left(w_{1:N}|\mu, \tau\right) p\left(\mu|\xi, \psi\right) \propto \text{Normal}\left(\mu; \frac{\bar{w}N\tau + \xi/\psi}{N\tau + 1/\psi}, \frac{1}{N\tau + 1/\psi}\right)$$

$$p\left(\tau|\mu, w_{1:N}\right) \propto p\left(w_{1:N}|\mu, \tau\right) p\left(\tau|\alpha, \beta\right) \propto \text{Gamma}\left(\tau; \alpha + \frac{N}{2}, \beta + \frac{1}{2}\bar{s}\right)$$

where $\bar{w} = \sum_n w_n$ and $\bar{s} = \frac{1}{N}\sum_n (w_n - \bar{w})^2$. Thus, the updates in each iteration of the Gibbs sampler require the simulation of one normal and one gamma random variable

$$\mu^{(j+1)} \sim \text{Normal}\left(\frac{\bar{w}N\tau^{(j)} + \xi/\psi}{N\tau^{(j)} + 1/\psi}, \frac{1}{N\tau^{(j)} + 1/\psi}\right)$$
$$\tau^{(j+1)} \sim \text{Gamma}\left(\alpha + \frac{N}{2}, \beta + \frac{1}{2}\bar{s}\right).$$

Example 5.10: Gibbs sampling for a Bayesian Gaussian mixture

As an alternative example, consider a Normal(μ_1, σ^2) generating observations with probability ω_1 and a Normal(μ_2, σ^2) generating observations with probability $\omega_2 = 1 - \omega_1$. Suppose that we obtain observations w_n, for $n = 1, \ldots, N$, through this model and our task is to estimate: (i) the locations μ_1, μ_2 of the two Normal distributions; (ii) their spread σ; and (iii) which of the two normals generated each observation w_n.

To estimate μ_1 and μ_2 for task (i) we can use a common Normal(ξ, ψ) prior. To estimate σ for task (ii), we may reparametrize the Normal distributions in terms of precision $\tau = 1/\sigma^2$ and subsequently place a Gamma(α, β) prior on τ. Finally, for each observation w_n, we can consider an indicator variable s_n, which is 1 if w_n were generated by Normal$(\mu_1, 1/\tau)$ or 2 if w_n were generated by Normal$(\mu_2, 1/\tau)$, so for task (iii) we simply need to estimate the values of s_n.

With this set-up the entire model is

$$\begin{aligned}
\tau &\sim \text{Gamma}(\alpha, \beta) \\
\mu_1 &\sim \text{Normal}(\xi, \psi) \\
\mu_2 &\sim \text{Normal}(\xi, \psi) \\
s_n &\sim \text{Categorical}_{1,2}(\omega_1, \omega_2) \\
w_n | s_n, \mu_1, \mu_2, \tau &\sim \text{Normal}\left(\mu_{s_n}, \frac{1}{\tau}\right), \qquad n = 1, \ldots, N.
\end{aligned}$$

This model is described by $(\tau, \mu_1, \mu_2, s_{1:N})$. To characterize the posterior $p(\tau, \mu_1, \mu_2, s_{1:N} | w_{1:N})$, we may compute samples $\tau^{(j)}$, $\mu_1^{(j)}$, $\mu_2^{(j)}$, $s_{1:N}^{(j)}$ through a Gibbs sampler. For concreteness, we work each conditional explicitly

- Generate $\tau^{(j)}$ by sampling from

$$\begin{aligned}
&p(\tau | \mu_1^{(j-1)}, \mu_2^{(j-1)}, s_{1:N}^{(j-1)}, w_{1:N}) \\
&\propto p(\tau | \mu_1^{(j-1)}, \mu_2^{(j-1)}, s_{1:N}^{(j-1)}) p(w_{1:N} | \tau, \mu_1^{(j-1)}, \mu_2^{(j-1)}, s_{1:N}^{(j-1)}) \\
&= p(\tau) \prod_{n=1}^{N} p(w_n | \tau, \mu_1^{(j-1)}, \mu_2^{(j-1)}, s_n^{(j-1)}) \\
&= \text{Gamma}(\tau; \alpha, \beta) \prod_{n=1}^{N} \text{Normal}\left(w_n; \mu_{s_n^{(j-1)}}^{(j-1)}, \frac{1}{\tau^{(j-1)}}\right) \\
&\propto \text{Gamma}\left(\tau; \alpha + \frac{N}{2}, \beta + \frac{1}{2}\sum_{n=1}^{N} (w_n - \mu_{s_n^{(j-1)}}^{(j-1)})^2\right).
\end{aligned}$$

- Generate $\mu_1^{(j)}$ by sampling from

$$\begin{aligned}
&p(\mu_1 | \tau^{(j)}, \mu_2^{(j-1)}, s_{1:N}^{(j-1)}, w_{1:N}) \\
&\propto p(\mu_1 | \tau^{(j)}, \mu_2^{(j-1)}, s_{1:N}^{(j-1)}) p(w_{1:N} | \tau^{(j)}, \mu_1, \mu_2^{(j-1)}, s_{1:N}^{(j-1)}) \\
&= p(\mu_1) \prod_{n=1}^{N} p(w_n | \tau^{(j)}, \mu_1, \mu_2^{(j-1)}, s_n^{(j-1)})
\end{aligned}$$

$$
\begin{aligned}
&\propto p(\mu_1) \prod_{s_n^{(j-1)}=1} p(w_n|\tau^{(j)}, \mu_1) \\
&= \text{Normal}(\mu_1; \xi, \psi) \prod_{s_n^{(j-1)}=1} \text{Normal}\left(w_n; \mu_1, \frac{1}{\tau^{(j)}}\right) \\
&\propto \text{Normal}\left(\mu_1; \frac{\bar{w}_1^{(j-1)} N_1^{(j-1)} \tau^{(j)} + \xi/\psi}{N_1^{(j-1)}\tau^{(j)} + 1/\psi}, \frac{1}{N_1^{(j-1)}\tau^{(j)} + 1/\psi}\right)
\end{aligned}
$$

where $N_1^{(j-1)}$ is the number of 1 in $s_{1:N}^{(j-1)}$ and $\bar{w}_1^{(j-1)}$ is the mean value of the associated observations.

- Generate $\mu_2^{(j)}$ by sampling from

$$
\begin{aligned}
&p(\mu_2|\tau^{(j)}, \mu_1^{(j)}, s_{1:N}^{(j-1)}, w_{1:N}) \\
&\propto p(\mu_2|\tau^{(j)}, \mu_1^{(j)}, s_{1:N}^{(j-1)}) p(w_{1:N}|\tau^{(j)}, \mu_1^{(j)}, \mu_2, s_{1:N}^{(j-1)}) \\
&= p(\mu_2) \prod_{n=1}^{N} p(w_n|\tau^{(j)}, \mu_1^{(j)}, mu_2, s_n^{(j-1)}) \\
&\propto p(\mu_2) \prod_{s_n^{(j-1)}=2} p(w_n|\tau^{(j)}, \mu_2) \\
&= \text{Normal}(\mu_2; \xi, \psi) \prod_{s_n^{(j-1)}=2} \text{Normal}\left(w_n; \mu_2, \frac{1}{\tau^{(j)}}\right) \\
&\propto \text{Normal}\left(\mu_2; \frac{\bar{w}_2^{(j-1)} N_2^{(j-1)} \tau^{(j)} + \xi/\psi}{N_2^{(j-1)}\tau^{(j)} + 1/\psi}, \frac{1}{N_2^{(j-1)}\tau^{(j)} + 1/\psi}\right)
\end{aligned}
$$

where $N_2^{(j-1)}$ is the number of 2 in $s^{(j-1)}$ and $\bar{w}_2^{(j-1)}$ is the mean value of the associated observations.

- Generate $s_{1:N}^{(j)}$ by sampling each indicator s_n individually from

$$
\begin{aligned}
&p(s_n|\tau^{(j)}, \mu_1^{(j)}, \mu_2^{(j)}, s_{1:n-1}^{(j)}, s_{n+1:N}^{(j-1)}, w_{1:N}) \\
&\propto p(w_n|s_n, \tau^{(j)}, \mu_1^{(j)}, \mu_2^{(j)}, s_{1:n-1}^{(j)}, s_{n+1:N}^{(j-1)}, w_{1:n-1}, w_{n+1:N}) \\
&\times p(s_n|\tau^{(j)}, \mu_1^{(j)}, \mu_2^{(j)}, s_{1:n-1}^{(j)}, s_{n+1:N}^{(j-1)}, w_{1:n-1}, w_{n+1:N}) \\
&= p(w_n|s_n, \tau^{(j)}, \mu_1^{(j)}, \mu_2^{(j)}) p(s_n) \\
&= \text{Categorical}_{1,2}\left(s_n; \tilde{\omega}_1, \tilde{\omega}_2\right)
\end{aligned}
$$

where $\tilde{\omega}_1 = Z^{-1}\omega_1 \text{Normal}\left(w_n; \mu_1^{(j)}, \frac{1}{\tau^{(j)}}\right)$, $\tilde{\omega}_2 = Z^{-1}\omega_2 \text{Normal}\left(w_n; \mu_2^{(j)}, \frac{1}{\tau^{(j)}}\right)$, and $Z = \omega_1 \text{Normal}\left(w_n; \mu_1^{(j)}, \frac{1}{\tau^{(j)}}\right) + \omega_2 \text{Normal}\left(w_n; \mu_2^{(j)}, \frac{1}{\tau^{(j)}}\right)$.

Within-Gibbs samplings schemes

When the random variable of interest $\boldsymbol{R}$ is part of a complex problem, there is often a natural way in which to group its components $R_{1:M}$. For instance, as we saw in example 5.10 variables partitioned in three groups: tags in $R_1 = s_{1:N}$; locations in $R_2 = (\mu_1, \mu_2)$; and precision in $R_3 = \tau$. Such problem-motivated groupings often lead to intuitive Gibbs schemes. Nevertheless, in the modeling of physical systems, one or more of the full conditionals in algorithms 5.2 and 5.3 may be difficult or otherwise impossible to sample from. In this case, any draw from the full conditionals may be replaced by a Metropolis-Hastings iteration or any of its variants.

Example 5.11: Metropolis-Hastings within-Gibbs

As a concrete case, consider the same setup as in example 5.7. In this case, the model variable $\boldsymbol{R} = (R_1, R_2, R_3)$ consists of of 3 components, with target and full conditionals given by

$$\begin{aligned}\pi(r_1, r_2, r_3) &= p(r_1, r_2, r_3)\\ \pi^1_{r_2,r_3}(r_1) &= p(r_1|r_2, r_3)\\ \pi^2_{r_1,r_3}(r_2) &= p(r_2|r_1, r_3)\\ \pi^3_{r_1,r_2}(r_3) &= p(r_3|r_1, r_2).\end{aligned}$$

As we have already seen, to advance from $\boldsymbol{r}^{\text{old}}$ to $\boldsymbol{r}^{\text{new}}$, a fixed sweep Gibbs sampler requires drawing successive samples

$$\begin{aligned}R_1^{\text{new}}|r_2^{\text{old}}, r_3^{\text{old}} &\sim \Pi^1_{r_2^{\text{old}}, r_3^{\text{old}}}\\ R_2^{\text{new}}|r_1^{\text{new}}, r_3^{\text{old}} &\sim \Pi^1_{r_1^{\text{new}}, r_3^{\text{old}}}\\ R_3^{\text{new}}|r_1^{\text{new}}, r_2^{\text{new}} &\sim \Pi^1_{r_1^{\text{new}}, r_2^{\text{new}}}.\end{aligned}$$

Suppose that the second conditional $\Pi^2_{r_1^{\text{new}}, r_3^{\text{old}}}$ cannot be simulated, so we cannot obtain r_2^{new} directly. In this case, we may proceed with a Metropolis-Hastings step. In particular, to obtain r_2^{new}, we can first sample

$$r_2^{\text{prop}} \sim \mathbb{Q}_{r_1^{\text{new}}, r_2^{\text{old}}, r_3^{\text{old}}}$$

using a proposal distribution that generally may depend on r_2^{old} and potentially on the other variables as well. The acceptance ratio of this step is

$$A_{r_2^{\text{old}}}(r_2^{\text{prop}}) = \frac{\pi^2_{r_1^{\text{new}}, r_3^{\text{old}}}(r_2^{\text{prop}})}{\pi^2_{r_1^{\text{new}}, r_3^{\text{old}}}(r_2^{\text{old}})} \frac{Q_{r_1^{\text{new}}, r_2^{\text{prop}}, r_3^{\text{old}}}(r_2^{\text{old}})}{Q_{r_1^{\text{new}}, r_2^{\text{old}}, r_3^{\text{old}}}(r_2^{\text{prop}})}$$

and the acceptance test becomes

- If $u < A_{r_2^{\text{old}}}(r_2^{\text{prop}})$, the proposal is accepted and so $r_2^{\text{new}} = r_2^{\text{prop}}$
- If $u \geq A_{r_2^{\text{old}}}(r_2^{\text{prop}})$, the proposal is rejected and so $r_2^{\text{new}} = r_2^{\text{old}}$

where $u \sim \text{Uniform}_{[0,1]}$, as usual.

This strategy of obtaining Gibbs samplers is termed *Metropolis-Hastings within-Gibbs* or *Metropolis within-Gibbs* depending upon the type of proposal used in the indirect sampling steps and the form of the acceptance ratio used. Although the resulting schemes are valid and often easy to implement, their performance is generally poor. This is because rejections of the proposed samples lead to highly correlated MCMC chains that require an exuberantly large number of repetitions in order to adequately characterize a given target. To improve such a scheme, the indirect sampling steps may be iterated several times before proceeding. In this way, the chances of generating at least one proposal that will be accepted increase allowing the chain to proceed further. Provided the indirect sampling steps involve little computational cost, this is a viable solution for sampling complex targets.

5.3 Processing and interpretation of MCMC

As we saw earlier, the common MC or MCMC task is to obtain samples $r^{(1)}, r^{(2)}, \cdots, r^{(J)}$ from a target $\pi(r)$ which is otherwise difficult to characterize. When these samples are *independent*, such as when they are generated by running an MC scheme, we can directly use them in empirical approximations like eq. (5.2). As the total number of samples J that we use increases, the approximation improves and, in the long run, empirical averages like $\frac{1}{J}\sum_{j=1}^{J} g(r^{(j)})$ converge to their exact counterparts $\langle g(r)\rangle$. Practically, this means that we can make an MC approximation as accurate as necessary by simply computing additional samples $r^{(j)}$. Therefore, as $J \to \infty$, our approximation becomes insignificant.

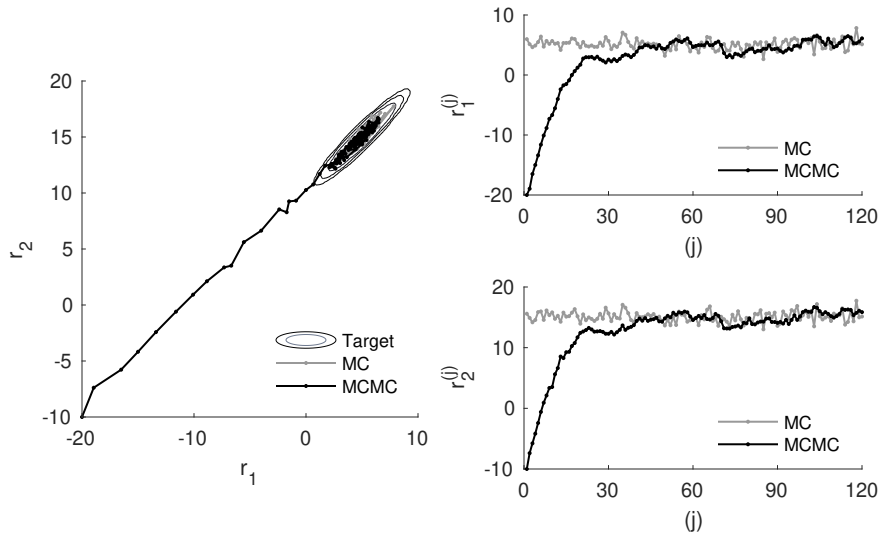


Figure 5.9: MCMC samples from a highly correlated bivariate target. On the left we show the target and the generated chain. On the right, we show how the evolution of the individual components over the course of the chain. For sake of comparison, we superimpose also a sequence of MC samples.

However, running an MCMC scheme yields samples that *depend* upon each other. For this reason, empirical approximations like eq. (5.2) are generally biased and the bias depends on how strongly samples depend on one another. Fortunately, given an MCMC chain, there is a sequence of standard procedures that we can follow to reduce such dependencies and, essentially, recover nearly independent MC samples.

Here, we describe some practical techniques to minimize biases introduced by MCMC sampling and considerably improve the accuracy of our approximations. We start by a motivating example and then proceed with a more formal discussion.

Example 5.12: Mixing of MC and MCMC

We consider a Gibbs sampler for a bivariate target consisting of correlated Normal components $\boldsymbol{r} = (r_1, r_2)$. Specifically, the target density is

$$\pi(r_1, r_2) \propto \exp\left(-\frac{1}{2(1-\rho^2)}\left(\left(\frac{r_1-\mu_1}{\sigma_1}\right)^2 + \left(\frac{r_2-\mu_2}{\sigma_2}\right)^2 - \left(2\rho\frac{(r_1-\mu_1)(r_2-\mu_2)}{\sigma_1\sigma_2}\right)\right)\right)$$

where μ_1, μ_2 are the means, σ_1, σ_2 are the standard deviations, and ρ is the correlation coefficient.

Figure 5.9 shows an MCMC chain targeting $\pi(r_1, r_2)$. For illustrative purposes, we use $\mu_1 = 5$, $\mu_2 = 15$, $\sigma_1 = 1$, $\sigma_2 = 1$ and a high correlation coefficient $\rho = 0.95$. As this target has a standard form, we can also draw MC samples from it and, for comparison, in fig. 5.9 we also show an MC sequence.

Visual inspection of the two sampled sequences reveals three important characteristics:

1. In the long run, the MC and MCMC sequences intermingle, indicating that they both produce samples with the same statistics.
2. Unlike the MC sequence which reproduces the targeted statistics immediately, the MCMC sequence reproduces the targeted statistics only after an initial period covering the first ≈ 40 samples.
3. Unlike the MC sequence where successive samples are uncorrelated, MCMC samples are profoundly correlated.

These characteristics are common to MCMC chains and apply universally to the output of any MCMC method, no matter how the initial sample in the chain is selected or the sampler used to advance from sample to sample. For example, a similar behavior was observed in figs. 5.3 and 5.4 that we encountered earlier in this chapter.

These characteristics determine how well our MCMC chain resembles independent samples. Accordingly, they impact the approximations derived based on $r^{(j)}$ and for this reason they determine the overall quality of the MCMC scheme used. For instance, a chain that does not reproduce the target's statistics is useless; while, a chain that reproduces the target's statistics quickly is preferable to a chain that does so after a long period. Similarly, a chain whose successive samples are almost uncorrelated is preferable to a chain whose successive samples are correlated.

Practically, such features are influenced by the sampler used and tend to be related to each other. Typically, a good sampler performs well in all three respects and for this reason allows for reliable empirical approximations derived from chains of only small size J. When this happens, we may colloquially say that a sampling scheme mixes well.

Of course, mixing is a qualitative property for which the golden standard is set by MC schemes that, due to the generation of independent samples, achieve ideal mixing. By contrast, the dependence among samples in an MCMC scheme degrade their quality and so their mixing is less, often considerably less, than ideal. Nevertheless, as we mentioned above, given an MCMC chain, we can always improve its mixing characteristics. We will see that this entails discarding samples from the chain which, at first, indicates a reduction in efficiency; however, discarding an optimal number of samples nearly always leads to dramatically improved approximations.

More formally, given an MCMC chain $r^{(0)} \to r^{(1)} \to \cdots \to r^{(J)}$, before we use its samples to derive approximations, we need to address three attributes:

1. Identify whether the chain has fully explored the target and started reproducing its statistics.
2. Determine the initial period until the chain starts reproducing the targeted statistics.
3. Determine the lag between consecutive samples that can be considered almost independent.

Answering these questions is more or less achieved by inspection of the appropriate plots such as those shown in fig. 5.9. These are termed *trace plots* and depict how samples $r^{(j)}$ evolve over the course of the chain.

5.3.1 Assessing convergence

Assessing whether an MCMC chain reproduces the targeted statistics or not is a qualitative problem. Practically, we assess whether the chain has converged to the target by inspection of the trace plots. The critical characteristic is for the chain to revisit regions already sampled rather than to keep discovering new regions and never passing near past samples again. For example, the chain depicted in panel A of fig. 5.4 jumps from one region to the other once and never returns. This is an indication that the chain is still evolving and, when this scenario is encountered, additional samples should be produced. Ideally, once a chain converges to the target, its trace plots fluctuate around similar values and sporadically jump back and forth between potentially separated regions. For example, the chain depicted in panel B of fig. 5.4 jumps in and out of each mode multiple times indicating that its statistics have stabilized.

5.3.2 Burn-in removal

Once a convergent MCMC has been identified, we need to determine the number of initial samples that it takes until convergence is achieved. Although this process is more quantitive than the assessment of convergence in the first place, it still relies on trace plots. This period can be found by pinpointing the minimum number of samples in the initial portion of the chain that need to be discarded in order for the chain's statistics to stop changing.

This phase is termed *burn-in* or *warm-up* and can be more accurately located by comparing sample statistics across successive batches. For example, fig. 5.10 illustrates how the sample mean computed over three successive batches changes over the MCMC chain of fig. 5.9. As can be seen, the mean in both trace-plots over the first batch differs from the means over the second and third batches. This indicates that, for this particular chain, burn-in extends over the first batch.

In any case, the chain statistics stabilize only after burn-in, so the samples identified to belong in this phase should be removed from the chain and only the remaining samples should be used for further processing. So, from the original chain

$$r^{(0)} \to \cdots \to r^{(j-1)} \to r^{(j)} \to r^{(j+1)} \to \cdots \to r^{(J)},$$

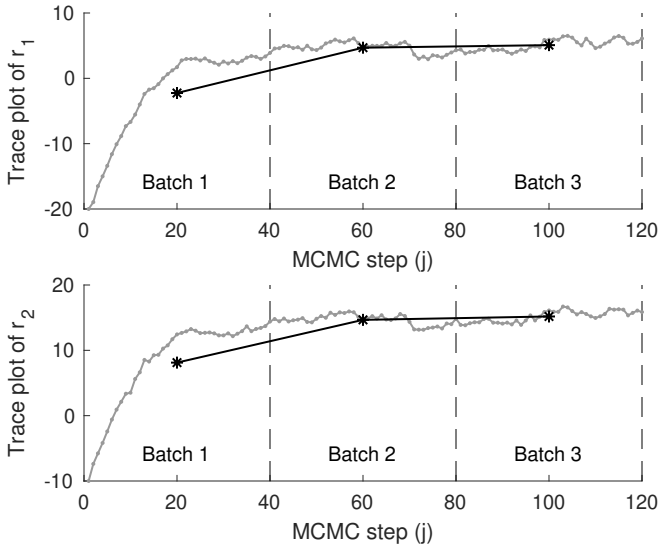


Figure 5.10: Trace plots and corresponding batch means of the MCMC chain depicted in fig. 5.9. As can be seen, batch statistics stabilize only after the first ≈ 40 samples.

after burn-in removal, we end up with a shorter chain

$$r^{(j_{\min})} \to \cdots \to r^{(j-1)} \to r^{(j)} \to r^{(j+1)} \to \cdots \to r^{(J)},$$

where $j_{\min}$ marks the end of the warm-up phase.

5.3.3 Thinning

Finally, once we have identified a convergent MCMC chain and removed burn-in, we need to quantify the correlation between the remaining samples. As such correlations depend on the lag d separating two samples $r^{(j)}$ and $r^{(j+d)}$, this is achieved by the *autocorrelation* along the course of the chain

$$\rho_d = \frac{\sum_{j=j_{\min}}^{J-j_{\min}+1-d} \left(r^{(j)} - \bar{r}\right)\left(r^{(j+d)} - \bar{r}\right)}{\sum_{j=j_{\min}}^{J-j_{\min}+1} \left(r^{(j)} - \bar{r}\right)^2}, \qquad d = 0, 1, \ldots, J - j_{\min}$$

where the mean is computed by $\bar{r} = \frac{1}{J-j_{\min}+1} \sum_{j=j_{\min}}^{J} r^{(j)}$.

As can be seen in fig. 5.11, the autocorrelation ρ_d quantifies how tightly related successive samples are in the MCMC chain.Typically, the autocorrelation is high at small lags and decreases in a nearly exponential fashion at larger lags. Accordingly, a lag such that $\rho_d \approx 0$ is a good indication of how many samples need to be generated until our MCMC chain resembles independent sampling. In practice, of course, it is sufficient to set a lower threshold $\rho_{\min} \approx 10\%$ and consider the minimum lag $d_{\min}$ that $\rho_{d_{\min}}$ drops below this threshold.

Once $d_{\min}$ is found, a downsampling process termed *thinning*, in which we maintain only 1 out of every $d_{\min}$ samples from our MCMC chain, can be used to recover independent samples. In summary, from the chain that remains after burn-in removal

$$r^{(j_{\min})} \to \cdots \to r^{(j-1)} \to r^{(j)} \to r^{(j+1)} \to \cdots \to r^{(J)}$$

we end up with a thinned chain

$$r^{(j_1)} \to \cdots \to r^{(j_{i-1})} \to r^{(j_i)} \to r^{(j_{i+1})} \to \cdots \to r^{(j_I)}$$

where $j_i - j_{i-1} = d_{\min}$. The MCMC samples $r^{(j_i)}$ remaining in the thinned chain resample nearly iid samples drawn from our target and, essentially, can be used as they had been generated by an MC scheme.

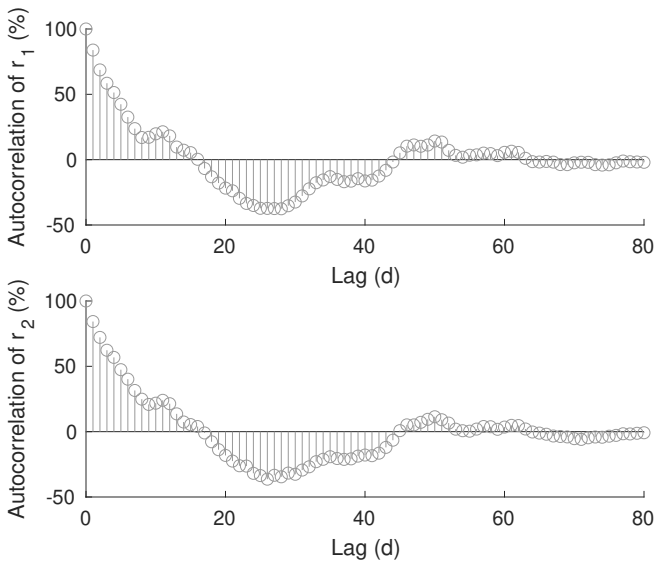


Figure 5.11: Autocorrelation of the MCMC chain depicted in fig. 5.10.

As shown in fig. 5.12, following the procedure above, we can turn a given chain of MCMC samples $r^{(j)}$ into MC ones $r^{(j_i)}$ on which we can safely rely to derive empirical approximations like eq. (5.2).

5.4 Exercise problems

Exercise 5.1: MC sampling

In the same context as in exercise 1.12, assume that the iid random variables R_1, R_2, R_3 follow Uniform$_{[-1,+1]}$ and develop a Monte Carlo method to estimate the probability of the polynomial $r_1 x^2 + r_2 x + r_3$ having real roots.

Exercise 5.2: MC sampling of prior and posterior

In the context of example 5.1, generate MC samples to characterize both the prior and posterior probability distributions of the parameters. Specifically, use the generated samples to create histograms as well as to compute mean values.

Exercise 5.3: Normalization

Recover the normalized form $\pi(r)$ of the unnormalized target $\bar{\pi}(r)$ of example 5.2. Show analytically that $\bar{\pi}(r) = 1$ is unnormalizable over the entire real line and that $\bar{\pi}(r) = r^{-1}(1-r)^{-1}$ is unnormalizable over the interval between 0 and 1.

Exercise 5.4: A Metropolis-Hastings sampler for a truncated Normal target

Develop a Metropolis-Hastings sampler to generate samples from a Normal random variable that is truncated between $r_{\min}$ and $r_{\max}$. Use Normal and Beta proposals. For the latter proposal, use a translation and a stretching such that the proposed values fall between $r_{\min}$ and $r_{\max}$.

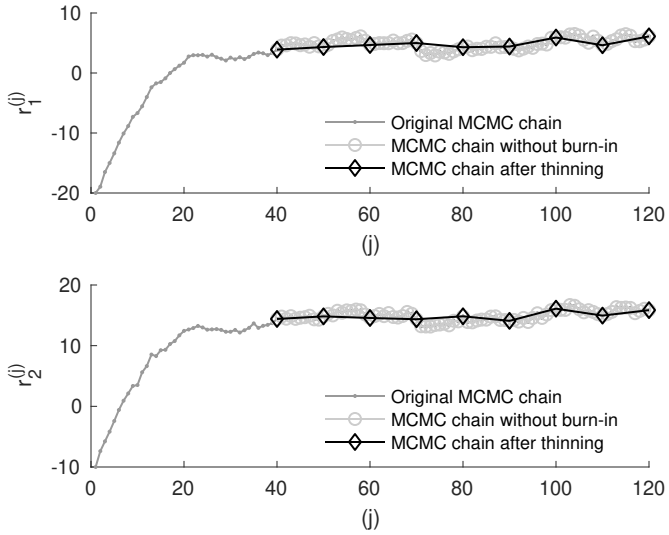


Figure 5.12: Recovering MC samples from a given MCMC chain is achieved by discarding burn-in samples and thinning. Here, we show nearly MC samples $r^{(r_i)}$ resulting from the chain depicted in fig. 5.9.

Exercise 5.5: Barker's acceptance

Verify that the Metropolis-Hastings sampler remains valid even if the acceptance ratio on eq. (5.9) is replaced by *Barker's ratio*

$$A_{r^{\text{old}}}\left(r^{\text{prop}}\right) = \left(1 + \frac{\pi(r^{\text{old}})Q_{r^{\text{old}}}\left(r^{\text{prop}}\right)}{\pi(r^{\text{prop}})Q_{r^{\text{prop}}}\left(r^{\text{old}}\right)}\right)^{-1}.$$

Exercise 5.6: Acceptance probabilities

Verify the identity in note 5.6.

Exercise 5.7: A Metropolis sampler for Cauchy and Laplace targets

Develop a Metropolis sampler to generate samples from a Cauchy random variable and a Laplace random variable. Use Normal and StudentT proposals.

Exercise 5.8: Detailed balance in Gibbs sampling[a]

[a] This is an advanced topic and could be skipped on a first reading.

Consider a tri-variate target $\pi(r_1, r_2, r_2)$.

1. Show analytically that the *random* sweep Gibbs sampler fulfills the detailed balance condition.
2. Use an example of a toy target $\pi(r_1, r_2, r_2)$, to show analytically that the *fixed* sweep Gibbs sampler does not, in general, fulfill the detailed balance condition.
3. Show analytically that a variant of the *fixed sweep* Gibbs sampler, where each iteration involves 5 updates:
   - Update r_1 by sampling from $r_1|r_2, r_3$
   - Update r_2 by sampling from $r_2|r_1, r_3$
   - Update r_3 by sampling from $r_3|r_1, r_2$
   - Update r_2 by sampling from $r_2|r_1, r_3$
   - Update r_1 by sampling from $r_1|r_2, r_3$

   fulfills the detailed balance condition.

Exercise 5.9: Verification

Verify the formulas for the acceptance ratios in examples 5.2 and 5.3 and the full conditionals in example 5.9.

Exercise 5.10: Ergodicity recovery of the Gibbs sampler

For the target $\pi(r_1, r_2)$ provided in example 5.8, consider the following transformation of variables

$$w_1 = \frac{r_1 + r_2}{2}, \qquad w_2 = \frac{r_1 - r_2}{2}.$$

Develop a Gibbs sampler for the transformed random variables and show, analytically or computationally, that this sampler is ergodic.

Exercise 5.11: Gibbs sampler for mixture models

In example 5.10 we developed a Gibbs sampler for a mixture model of a Normal$(\mu_1, 1/\tau)$ and a Normal$(\mu_2, 1/\tau)$. In the example, we assumed that the observations $w_{1:N}$ stem from the first and second components with known probabilities $\omega_1 = \omega$ and $\omega_2 = 1 - \omega$, respectively. However, generally, the value of ω may be unknown and so we have to estimate it just like any other variable.
Place a Beta prior on ω and develop a Gibbs sampler to draw posterior samples from $p(\omega, \tau, \mu_1, \mu_2, s_{1:N}|w_{1:N})$. In this setup, the entire model is

$$\begin{aligned}
\omega &\sim \text{Beta}(A, B) \\
\tau &\sim \text{Gamma}(\alpha, \beta) \\
\mu_1 &\sim \text{Normal}(\xi, \psi) \\
\mu_2 &\sim \text{Normal}(\xi, \psi) \\
s_n|\omega &\sim \text{Categorical}_{1,2}(\omega, 1-\omega) \\
w_n|s_n, \mu_1, \mu_2, \tau &\sim \text{Normal}\left(\mu_{s_n}, \frac{1}{\tau}\right), \qquad n = 1, \ldots, N
\end{aligned}$$

Exercise 5.12: A Gibbs sampler for Binomial likelihoods[a]

[a] This is an advanced topic and could be skipped on a first reading.

For each n, consider the pairs of random variables

$$r_n^1 \sim \text{Binomial}\left(m_n^1, p\right), \qquad r_n^2 \sim \text{Binomial}\left(m_n^2, q\right).$$

Suppose we observe three pairs with the following values

n	m_n^1	m_n^2	$r_n^1 + r_n^2$
1	5	5	7
2	6	4	5
3	4	6	6

Place uniform priors on p and q, and develop a Gibbs sampler to obtain posterior samples. Use the generated samples to estimate the values of p and q.

Exercise 5.13: A hyper-hyper-model for Normal likelihoods

In example 5.9 we applied independent Normal-Gamma priors to estimate the center and spread of an underlying Normal distribution. In doing so, we used known values for the hyperparameters ξ, ψ, α, β; however, in many practical applications specifying values for these hyperparameters might not be easy. In such cases, we may apply

a hyper-hyper-model

$$\begin{aligned}
\xi &\sim \text{Normal}(\eta, \zeta) \\
\psi &\sim \text{Gamma}(\kappa, \lambda) \\
\\
\alpha &\sim \text{Gamma}(\gamma, \omega) \\
\beta &\sim \text{Gamma}(\rho, \phi) \\
\\
\mu|\xi, \psi &\sim \text{Normal}\,(\xi, \psi) \\
\tau|\alpha, \beta &\sim \text{Gamma}(\alpha, \beta) \\
\\
w_n|\mu, \tau &\sim \text{Normal}\left(\mu, \frac{1}{\tau}\right), \qquad\qquad n = 1, \ldots, N.
\end{aligned}$$

Assume the values of the hyper-hyper-parameters $\eta, \zeta, \kappa, \lambda, \gamma, \omega, \rho, \phi$ are given and develop a Metropolis-within-Gibbs scheme to sample from the joint posterior $p(\mu, \tau, \alpha, \beta, \xi, \psi|w_{1:N})$. In doing so, use Gibbs updates for $\mu, \tau, \beta, \xi, \psi$ and a Metropolis update for α.

Exercise 5.14: Interpretation of MCMC

Develop a Metropolis (additive) random walk to sample from a bivariate target $\boldsymbol{r} = (r_1, r_2)$ with the density

$$\pi(\boldsymbol{r}) \propto \exp\left(-10(r_1^2 - r_2)^2 - \left(r_2 - \frac{1}{4}\right)^4\right).$$

For the perturbations that define the random walk use

$$\boldsymbol{\epsilon}^{\text{prop}} \sim \text{Normal}_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$$

with three different scalings $\lambda = 0.01$, $\lambda = 1$, and $\lambda = 100$. For each case, initialize the chain at $(-1, -1)$ and generate 10^4 samples. Recover nearly MC samples for all cases.

Part II

Statistical models

Chapter 6

Mixture models

6.1 Introduction

6.2 Finite mixture models

6.3 EM for finite mixture problems

6.4 Bayesian finite mixture models

6.4.1 Priors for finite mixture models

6.4.2 A Gibbs sampler

6.5 Infinite mixture models

6.6 Latent feature models and Beta-Bernoulli processes

Let's rename this chapter to something like "Mixture and latent feature models"

(Sina will provide more here)

Here we envision allowing features of models, up to an infinite number of features, to turn off or on. We do this by creating "particles with loads".

For this reason, we introduce a finite, but large, model population consisting of N particles containing both active and inactive particles. These model particles are collectively indexed by $k = 1, 2, \ldots, K$. Estimating how many particles are actually warranted by the data under analysis is equivalent to estimating how many of those N particles are active, i.e. $b^k = 1$, while the remaining inactive ones, i.e. $b^k = 0$, have no impact whatsoever and are instantiated only for computational purposes.

To give a concrete example, the time at which the next death event occurs in a death process with death rate λ_d per particle depends on the number of active particles present currently. That is, the number of particles with active load 1, $t|\boldsymbol{b} = 1 \sim \text{Exp}(\sum_k b^k \lambda_d)$.

To ensure that each load b^k takes only values 0 or 1, we place a Bernoulli prior of weight q^k. In turn, on each weight q^k, we place a conjugate Beta hyperprior

$$q^k \sim \textbf{Beta}\,(A_k, B_k) \tag{6.1}$$

$$b^k|q^k \sim \textbf{Bernoulli}(q^k). \tag{6.2}$$

To ensure that the resulting formulation avoids overfitting, we make the specific choices $A_k = \alpha_k/K$ and $B_k = \beta_k(K-1)/K$. That is, the more particles are introduced, the smaller the probability of getting an active particle should become. Typically β_k is quite large.

Now, for the death process, we write

$$q^k \sim \textbf{Beta}\,(A_k, B_k) \qquad k = 1, 2, \ldots, K \tag{6.3}$$

$$b^k | q^k \sim \textbf{Bernoulli}(q^k) \qquad k = 1, 2, \ldots, K \tag{6.4}$$

$$\lambda_d | \gamma \sim \text{Gamma}(\gamma) \tag{6.5}$$

$$t | \boldsymbol{b} = 1, \lambda_d \sim \text{Exp}\left(\sum_k b^k \lambda_d\right) \tag{6.6}$$

6.7 Exercise problems

Exercise 6.1: Counting data

A Poisson mixture model for counting data. EM and Gibbs

Chapter 7

Gaussian processes

7.1 Gaussian process

Remaining plan for full chapter:
(discuss classification with GP involving nonparametric boundaries)

7.1.1 Motivating Gaussian Processes from simple regression

Here we present a Bayesian approach to non-linear regression. But, before we do so, we briefly discuss regression in broader terms.

Suppose we have data $\boldsymbol{y}_{1:N}$ expected to satisfy the linear relation The intercept and the slope of the linear relation can be thought of as two parameters, $\theta_0 = b$ and $\theta_1 = m$ respectively, that we wish to determine. Here x is the independent variable of the hypothetical relation $\mu(x) = mx + b$ for which we only have data for fixed values of x which we index i. In other words, to be more explicit, our data is provided in the form of N pairs $(x_i, y_i)_{i=1:N}$.

Assuming an observation model of

$$y_i = (mx_i + b) + \epsilon \tag{7.1}$$

$$\epsilon \sim \text{Normal}(0, \sigma^2) \tag{7.2}$$

the likelihood of the sequence of iid observations, assuming known σ^2, becomes

$$p(\boldsymbol{y}_{1:N}|m, b) = \prod_{i=1}^{N} p(y_i|m, b) \propto \prod_{i=1}^{N} e^{-\frac{1}{2\sigma^2}(y_i-(mx_i+b))^2}. \tag{7.3}$$

Maximum likelihood here reduces to the minimization of a χ^2, i.e., the minimization of $\sum_i (y_i - (mx_i + b))^2$ with respect to both m and b.

In principle, many values of the parameters m and b return an acceptable solution to this linear regression. That is, many choices of m and b will satisfy

$$\sum_i (y_i - (mx_i + b))^2 \approx N\sigma^2. \tag{7.4}$$

The range of acceptable parameter combinations only becomes greater as σ increases. The range also increases if we consider a polynomial relation $f(x) = \sum_{i=0}^{K-1} \theta_i x^i$ with parameters θ_i or any other non-linear $f(x)$ expanded in a basis set other than simple polynomials.

On the other hand, if we insist on minimizing the χ^2, we would then obtain fine-tuned values for the K parameters such that the χ^2 would eventually fall below $N\sigma^2$. This is a classic example of over-fitting.

To avoid over-fitting, we need an additional criterion, i.e., a penalty. An alternative is to maximize

$$\frac{1}{\sigma^2}\sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda\sum_{i=1}^{N}(f(x_i) - \mu(x_i))^2. \tag{7.5}$$

What this says is that we try to find the best parameters describing $f(x_i)$ for all i subject to the constraint that $f(x_i)$ follows $\mu(x_i)$ as tightly as constrained by λ, or more precisely the ratio of $\lambda/1/\sigma^2$.

In greater generality, we can think of dropping the index on x and think of $f(x)$ as some curve. In this case, maximizing eq. (7.5) over each $f(x_i)$ is equivalent to maximizing the logarithm of a posterior where the first term of eq. (7.5) is interpreted as a Gaussian likelihood on the continuous curve measured at $\mathbf{x}_{1:N}$ to give $\mathbf{y}_{1:N}$ and the second term is interpreted as a prior (over the continuous curve on which we place priors on each $\mathbf{x}_{1:N}$).

One shortcoming with this framework is that the posterior over $f(x)$ will be jagged (I should show a plot here for the special case of the diagonal kernel). We may want a smooth curve and, to achieve this, we can further generalize eq. (7.5) to include a spatial covariance between data points. That is,

$$\frac{1}{\sigma^2}\sum_{i=1}^{N}(y_i - f(x_i))^2 + \sum_{ij=1}^{N}(f(x_i) - \mu(x_i))K^{-1}(x_i, x_j)(f(x_j) - \mu(x_j)). \tag{7.6}$$

The particular choice of the covariance matrix, $\mathbf{K}$, can help smooth samples in the final posterior. (Julian says: It is not clear why this form leads to smoother curve. Better give a concrete example.)

7.1.2 Introduction to the Gaussian processes

Now we imagine a curve $f(x)$ defined over all values of x. This value of the curve can be evaluated at any point and, in particular, on any grid in x.

We say that the collection of such points, $f(x_i)$ say for $i = 1 : N$, is a Gaussian process if, for any choice of x_i, the probability distribution over $f(x_i)$ at those points is a multivariate Gaussian. The example below expands upon this definition.

Example 7.1: Definition of the Gaussian process

We consider a curve, $f(x)$, and imagine discretizing this curve on a grid. For pedagogical reasons, we consider $f(x)$ discretized on two grids and define $\{f(x_i)\}_{i=1:N}$ and $\{f(x_i)\}_{i=1:L}$. For now, we assume that the grids may interweave or overlap.

The multivariate according to which the f's on first grid are sampled is

$$(f(x_1), f(x_2), \cdots, f(x_N)) \sim \text{Normal}((\mu(x_1), \mu(x_2), \cdots, \mu(x_N)), \boldsymbol{K}) \tag{7.7}$$

$$\boldsymbol{f}_{1:N} \sim \text{Normal}(\boldsymbol{\mu}_{1:N}, \boldsymbol{K}) \tag{7.8}$$

where $\boldsymbol{K}$ is an $N \times N$ matrix whose matrix elements may, generally, depend on x_i, i.e., $K(x_i, x_j)$.

By the definition of the Gaussian process, the multivariate according to which the f's on both grids are sampled is

$$(\boldsymbol{f}_{1:N}, \boldsymbol{f}_{1:L}) \sim \text{Normal}\left(\begin{bmatrix}\boldsymbol{\mu}_{1:N}\\ \boldsymbol{\mu}_{1:L}\end{bmatrix}, \begin{bmatrix}\boldsymbol{K} & \boldsymbol{K}_*\\ \boldsymbol{K}_*^T & \boldsymbol{K}_{**}\end{bmatrix}\right). \tag{7.9}$$

where $\boldsymbol{K}$ is the same $N \times N$ matrix as before, $\boldsymbol{K}_{**}$ is an $L \times L$ matrix, $\boldsymbol{K}_*$ is an $N \times L$ matrix and $\boldsymbol{K}_*^T$ is $\boldsymbol{K}_*$'s transpose.

For now, we may assume that the matrix elements, $K(x_i, x_j)$, of all submatrices have the same form except that for the $\boldsymbol{K}_*$, say, the x_i's are taken from the first grid and the x_j's are taken from the second grid.

So, just to be clear, we call the collection of random variables $\{f(x_i)\}_{i=1:N}$ a Gaussian process. We call the distribution according to which $\{f(x_i)\}_{i=1:N}$ is sampled a multivariate Gaussian. Often, in the literature,

the distribution itself and the collection of random variables are both, interchangeably, called Gaussian processes. This is an abuse of language that we avoid here.

However, following accepted convention, if we treat the multivariate Gaussian from which $\{f(x_i)\}_{i=1:N}$ is sampled as a prior on $\{f(x_i)\}_{i=1:N}$, we will call this a *Gaussian process prior* and write

$$f(x) \sim GP(\mu(x), K(x, x')) \tag{7.10}$$

where $K(x, x')$ is often termed the Gaussian process kernel.

A common choice for $K(x, x')$ is the so-called squared exponential

$$K(x, x') = \tau^2 exp(-(x - x')^2/\ell^2) \tag{7.11}$$

and a common choice for $\mu(x)$ is 0.

The reason for the squared exponential is because the squared exponential introduces into the prior a preference for smooth curves (which become especially smooth as ℓ becomes larger and spans the grid size). The reason for the zero mean is because, as no information on data should be contained in the prior, the prior guess on the mean shape of the $f(x)$ curve is a flat line.

7.1.3 Sampling from the Gaussian process

So far we have only discussed the prior and not the data but it is worth discussing how one would sample a curve evaluated at fixed grid points from the Gaussian process prior. The idea here would be that – provided we are given a Gaussian observation noise model, conjugate to the Gaussian process – then sampling from the posterior would be equivalent to sampling from the prior albeit with updated means and kernels.

(In chapter 1 include problem on Box-Mueller method)

We begin with sampling of f from a univariate Normal (i.e., a simple Gaussian) with mean μ and variance σ^2 for which

$$f = \mu + \sigma\epsilon \tag{7.12}$$

$$\epsilon \sim \text{Normal}(0, 1). \tag{7.13}$$

The generalization to the multivariate normal shown below follows from eq. (7.13)

$$(f(x_1), f(x_2), \cdots, f(x_L)) = (\mu(x_1), \mu(x_2), \cdots, \mu(x_L)) + \boldsymbol{L} \cdot \boldsymbol{\epsilon} \tag{7.14}$$

$$\boldsymbol{\epsilon} \sim \text{Normal}(0, \mathbb{1}) \tag{7.15}$$

where $\boldsymbol{K} = \boldsymbol{L}\boldsymbol{L}^T$ and $\boldsymbol{L}$ is a lower triangular matrix obtained by Cholesky decomposition.

(Discuss Cholesky, discuss finite approximation of Cholesky using a finite basis set)

7.1.4 Gaussian process posterior

From the Gaussian process prior and an observation model with known variance per data point, σ^2, it is possible to write down the full joint distribution over both $\boldsymbol{f}_{1:L}$ and observations $\boldsymbol{y}_{1:N}$. As before, both grids are assumed not to have overlapping points.

The joint distribution over $\boldsymbol{f}_{1:L}$ and $\boldsymbol{y}_{1:N}$ is

$$(\boldsymbol{y}_{1:N}, \boldsymbol{f}_{1:L}) \sim \text{Normal}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\boldsymbol{K} + \sigma^2\mathbb{1} & \boldsymbol{K}_* \\ \boldsymbol{K}_*^T & \boldsymbol{K}_{**}\end{bmatrix}\right). \tag{7.16}$$

The posterior over $\boldsymbol{f}_{1:L}$ given data acquired $\boldsymbol{y}_{1:N}$, $p(\boldsymbol{f}_{1:L}|\boldsymbol{Y}_{1:N} = \boldsymbol{y}_{1:N}, \ell, \tau) = p(\boldsymbol{f}_{1:L}|\boldsymbol{y}_{1:N}, \ell, \tau)$, is then obtained from

$$p(\boldsymbol{f}_{1:L}|\boldsymbol{y}_{1:N}, \ell, \tau) = \frac{p(\boldsymbol{f}_{1:L}, \boldsymbol{y}_{1:N}|\ell, \tau)}{p(\boldsymbol{y}_{1:N}|\ell, \tau)} \propto p(\boldsymbol{f}_{1:L}, \boldsymbol{y}_{1:N}|\ell, \tau) \tag{7.17}$$

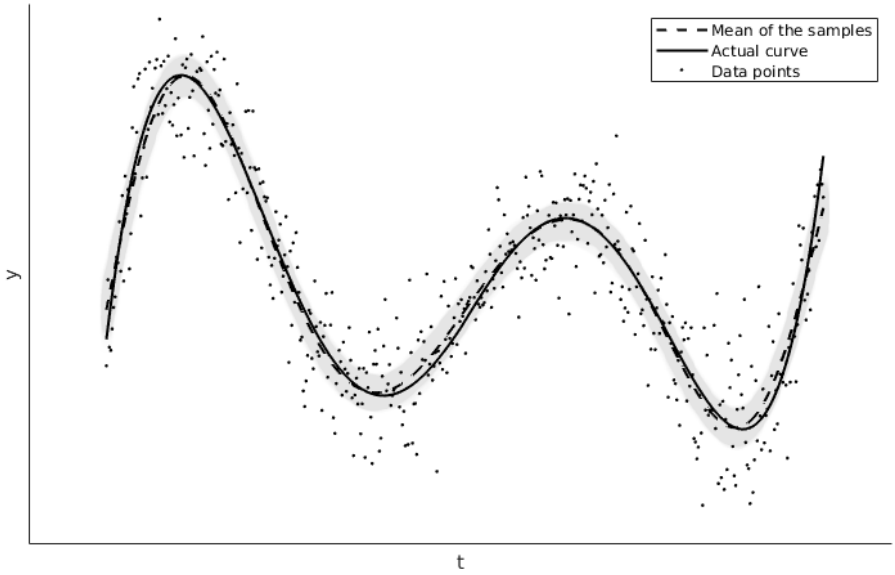


Figure 7.1: fill in.

where, plugging the form for the posterior from eq. (7.16) into eq. (7.17), we have

$$p(\boldsymbol{f}_{1:L}|\boldsymbol{y}_{1:N}, \ell, \tau) \propto p(\boldsymbol{f}_{1:L}, \boldsymbol{y}_{1:N}|\ell, \tau) \tag{7.18}$$

$$\propto \exp\left(-\frac{1}{2}\begin{bmatrix}\boldsymbol{y}_{1:N}^T & \boldsymbol{f}_{1:L}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{K}+\sigma^2\mathbb{1} & \boldsymbol{K}_* \\ \boldsymbol{K}_*^T & \boldsymbol{K}_{**}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{y}_{1:N} \\ \boldsymbol{f}_{1:L}\end{bmatrix}\right) \tag{7.19}$$

$$\propto \exp\left(-\frac{1}{2}\begin{bmatrix}\boldsymbol{y}_{1:N}^T & \boldsymbol{f}_{1:L}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{12}^T & \boldsymbol{A}_{22}\end{bmatrix}\begin{bmatrix}\boldsymbol{y}_{1:N} \\ \boldsymbol{f}_{1:L}\end{bmatrix}\right) \tag{7.20}$$

where $\boldsymbol{A}_{11} = ((\boldsymbol{K}+\sigma^2\mathbb{1}) - \boldsymbol{K}_*\boldsymbol{K}_{**}^{-1}\boldsymbol{K}_*^T)^{-1}$, $\boldsymbol{A}_{12} = \boldsymbol{A}_{21}^T = -(\boldsymbol{K}+\sigma^2\mathbb{1})^{-1}\boldsymbol{K}_*(\boldsymbol{K}_{**} - \boldsymbol{K}_*(\boldsymbol{K}+\sigma^2\mathbb{1})^{-1}\boldsymbol{K}_*^T)^{-1}$, and $\boldsymbol{A}_{22} = (\boldsymbol{K}_{**} - \boldsymbol{K}_*(\boldsymbol{K}+\sigma^2\mathbb{1})^{-1}\boldsymbol{K}_*^T)^{-1}$.

By completing the squares in $\boldsymbol{f}_{1:L}$, we can re-write eq. (7.20) in the form of a multivariate Gaussian

$$p(\boldsymbol{f}_{1:L}|\boldsymbol{y}_{1:N}, \ell, \tau) \propto \text{Normal}(\boldsymbol{f}_{1:L}; \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{K}}) \tag{7.21}$$

where

$$\tilde{\boldsymbol{\mu}} = \boldsymbol{K}_* \cdot (\boldsymbol{K}+\sigma^2\mathbb{1})^{-1} \cdot \boldsymbol{y}_{1:N} \tag{7.22}$$

$$\tilde{\boldsymbol{K}} = \boldsymbol{K}_{**} - \boldsymbol{K}_*^T \cdot (\boldsymbol{K}+\sigma^2\mathbb{1})^{-1} \cdot \boldsymbol{K}_*. \tag{7.23}$$

7.1.5 Boundary conditions and covariance functions

An important topic to discuss here are the boundaries of the Gaussian process.

Before we do so, we discuss Brownian motion as a prior model. In this case, consider a particle undergoing Brownian motion starting from the origin whose position after time t has elapsed is $x(t)$. Re-labeling x as f and t as x, from the solution to the diffusion equation, we have (re-writing 2.116 from Chapter 2) in the notation of this chapter

$$p(f(x)|D) = \frac{1}{(4\pi Dx)^{1/2}} e^{-\frac{f(x)^2}{4Dx}}. \tag{7.24}$$

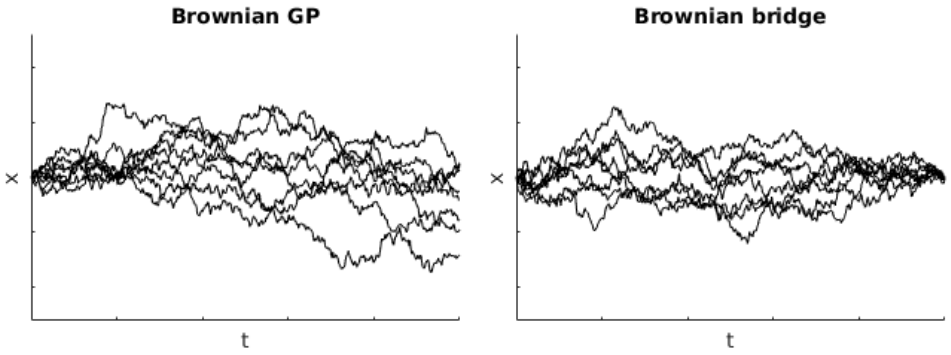


Figure 7.2: fill in.

Then, averaging over $p(f(x)|D)$, we find the mean particle displacement

$$\mu(x) = \langle f(x) \rangle = 0. \tag{7.25}$$

Likewise averaging over the joint distribution for two independent increments (where $u \geq x$), we have

$$K(x,u) = \langle f(x)f(u) \rangle = \langle f(x)f(x) \rangle + \langle f(x)(f(u) - f(x)) \rangle \tag{7.26}$$

$$= 2Dx. \tag{7.27}$$

Else for $u < x$, $K(x,u) = 2Du$.

Putting it all together, a Brownian Gaussian process prior is

$$\frac{f(x)}{\sqrt{2D}} \sim GP(\mu(x), K(x,x') = \mathsf{min}(x,x')). \tag{7.28}$$

Suppose now that both ends of the Brownian motion are fixed at $x = 0$ and $x = L$ to zero. We now define a new function, $\phi(x)$ such that

$$\phi(x) = f(x) - \frac{x}{L} f(L). \tag{7.29}$$

Here $\phi(x)$ satisfies these boundary conditions $\phi(0) = \phi(L) = 0$. As both $f(x)$ and $f(L)$ are Gaussian random variables, then $\phi(x)$, as the sum of two Gaussian random variables, is a Gaussian random variable as well. The mean of $\phi(x)$ taken with respect to $\phi(x)$ is therefore zero.

Next we can ask about the variance of $f(x) - \frac{x}{L} f(L)$ which is

$$\langle \phi(x)\phi(u) \rangle = \langle f(x)f(u) \rangle - \frac{x}{L} \langle f(L)f(u) \rangle - \frac{u}{L} \langle f(x)f(L) \rangle + \frac{xu}{L^2} \langle f^2(L) \rangle \tag{7.30}$$

$$= 2D \left(\mathsf{min}(x,u) - \frac{xu}{L} \right). \tag{7.31}$$

Putting it all together, a Brownian bridge Gaussian process prior tied down at $x = 0$ and $x = L$ is

$$\frac{\phi(x)}{\sqrt{2D}} \sim GP(0, K(x,x') = \mathsf{min}(x,u) - xu/L). \tag{7.32}$$

(include figures using a Gaussian likelihood with both normal Brownian GP and Brownian bridge)

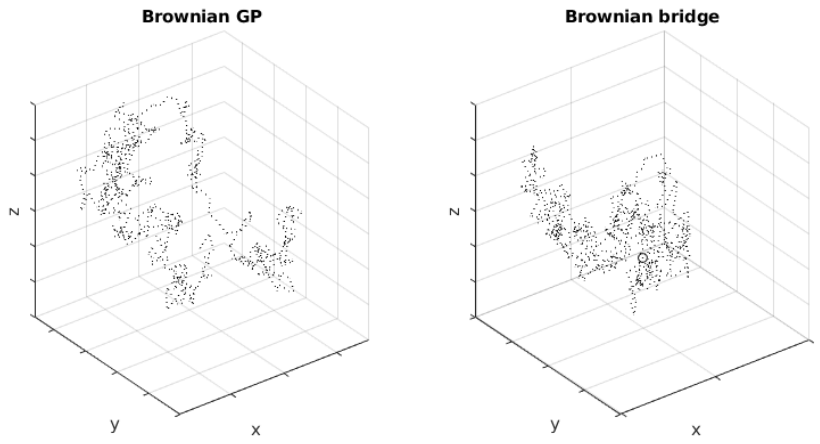


Figure 7.3: fill in.

7.1.6 Choice of covariance function

We've already seen three examples of covariance functions (one continuous and two discontinuous).

One key property that these–and all other– covariance functions must satisfy is that the covariance function, $K(x, x')$, must be non-negative when evaluated for any x and x'. In other words, the covariance matrix itself must be non-negative definite.

As such, any product or sum of two covariance functions is also a covariance function. Thus, taking the sum of different covariances for example can provide a way to capture short and long length scale fluctuations.

Through the covariance function we can introduce symmetries that we believe are important for the problem. For example, we can consider translational or scale invariance, namely $K(x, x') = K(x - x')$ and $K(x, x') = K(x/x')$, respectively.

7.1.7 Gaussian processes with uncertain input

So far, we have dealt with Gaussian processes as ways of nonparametrically fitting curves given a sequence $\mathbf{y}_{1:N}$ with some uncertainty σ^2.

In other words, we can think of having fixed time levels t with output $y(t)$ and our goal was to find a curve that fit these y's. Now suppose we have fixed time levels t with output $y(t)$. Suppose further that the ordered $y(t)$ is our input and the goal is to find the nonparametric distribution according to which they are sampled. That is, our input itself is uncertain (i.e., the y's are stochastic).

Concretely, here instead, we imagine an example where the $\mathbf{y}_{1:N}$ are themselves drawn from a probability distribution $P(y|f(\cdot), \cdot)$ or $P(y_i|y_{i-1}, f(\cdot), \cdot)$ that depends on some continuous function $f(\cdot)$ and other parameters. Our goal is to learn the shape of the continuous parameter dictating the shape of the distribution.

The measurements themselves may have uncertainty (just as we had considered before due to the observation process) however the problem here is unique in that we have an additional source of uncertainty on account of the y's being sampled from a probability distribution.

(Julian says: emission variable should not be same as latent state variable) As an example for the more general Markovian case with measurement noise, the likelihood reads

$$P(\mathbf{y}_{1:N}|f(\cdot), \boldsymbol{\theta}_o) = P(y_1|\boldsymbol{\theta}_o) \prod_{i=2}^{N} P(y_i|\boldsymbol{\theta}_o) P(y_i|y_{i-1}, f(\cdot)) \tag{7.33}$$

where $P(y_i|\boldsymbol{\theta}_o)$ is the emission distribution with observation parameters $\boldsymbol{\theta}_o$. In general, we would need to place priors on all unknowns. This includes a Gaussian process prior on $f(\cdot)$.

Unfortunately, we now run into the difficulty of non-conjugacy of our priors and likelihoods. In particular, depending on the precise form for $P(y_i|y_{i-1}, f(\cdot), \cdot)$, the conditional posterior $P(f(\cdot)|\mathbf{y}_{1:N}, \cdot)$ may not itself be Gaussian and eq. (7.23) no longer holds.

Before turning to questions of non-conjugacy, we envision an example where $P(y_i|y_{i-1}, f(\cdot), \cdot)$ is Gaussian in $f(\cdot)$. That is, we suppose $y_i = y_{i-1} + f(\cdot)$ and, therefore,

$$P(y_i|y_{i-1}, f(\cdot), \cdot) \propto \exp\left(-\frac{(y_i - y_{i-1} - f(\cdot))^2}{2\sigma^2}\right). \tag{7.34}$$

We can now place a Gaussian process prior on $f(\cdot)$ and we have a choice of mean and variance. For example, if we think of $f(\cdot)$ as related to a force in a Langevin equation, then one choice of mean is a linear force, $\alpha(y - y')$,

$$f(\cdot) \sim GP(\alpha(y - y'), K(y, y')). \tag{7.35}$$

(Ask Shep to add figures here)

7.1.8 Non-conjugate likelihoods with the Gaussian Process prior

There are deterministic methods like expectation propagation and Laplace methods. One idea is to use Laplace's method to approximate the mode of the posterior and keep pretending you are Gaussian.

(here we follow Titsias pretty closely)

These methods rely on factorizable likelihoods (see Titsias for this argument. I need to check it for myself...). They also only provide point estimates. If we want to sample the full posterior accurately, at least in the limit of a large number of samples, then we must use sampling methods.

Another method is to use MCMC. For example, we have from (cite from chapter 4)

$$R_{\boldsymbol{f}^{old}}(\boldsymbol{f}^{prop}) = \underbrace{\frac{p(\boldsymbol{y}_{1:N}|\boldsymbol{f}^{prop})p(\boldsymbol{f}^{prop})}{p(\boldsymbol{y}_{1:N}|\boldsymbol{f}^{old})p(\boldsymbol{f}^{old})}}_{\text{target}} \underbrace{\frac{Q_{\boldsymbol{f}^{prop}}(\boldsymbol{f}^{old})}{Q_{\boldsymbol{f}^{old}}(\boldsymbol{f}^{prop})}}_{\text{proposal}}. \tag{7.36}$$

The challenge here is finding a proposal distribution that is less prone to generating highly correlated samples. High correlation of samples is of particular concern for high dimensional inference problems.

(Julian says: Discussions in this page is particularly very difficult to understand. For example, why Eq.(5.37) leads to high acceptance ratio, and why does it lead to poor mixing? and so on.)

One choice of proposal distribution is the GP prior. This is advantageous as it has the same smoothness conditions as the function itself. But this leads to high rejections as it ignores the structure of the posterior. Another option is to use as mean of the prior, the previous sample iteration. Yet another is to propose an $\boldsymbol{f}$, $\boldsymbol{f}^{prop}$, by doing Gibbs sampling on the prior $p(f_i|\boldsymbol{f}_{-i})$ to avoid rejections caused by sampling the entire function. The challenge here is that samples may be tightly correlated and so it may be inefficient. The latter is called a Gibbs-like algorithm.

The goal is really to increase the acceptance rate and keep correlation between samples at a minimum by keeping computational cost low. One idea is to use a blocked Gibbs sampler. Here we partition the points $1:N$ (or 1:L?) into blocks indexed k. We use the GP prior $p(\boldsymbol{f}_k^{prop}|\boldsymbol{f}_{-k}^{old})$ as our proposal distribution. Then the acceptance ratio is then simply

$$R_{\boldsymbol{f}^{old}}(\boldsymbol{f}^{prop}) = \frac{p(\boldsymbol{y}_{1:N}|\boldsymbol{f}_k^{prop}, \boldsymbol{f}_{-k}^{old})}{p(\boldsymbol{y}_{1:N}|\boldsymbol{f}_k^{old}, \boldsymbol{f}_{-k}^{old})}. \tag{7.37}$$

We note that since the prior is identical to the proposal distribution, these cancel out in acceptance ratio. The basic idea is that you want to have groups of the appropriate size to keep the acceptance ratio pretty high. This can still result in poor mixing as variables between clusters are decoupled (by construction). As such, the variance of the proposal distribution is smaller in regions between blocks and the size of blocks can be adjusted to minimize this problem.

A more sophisticated way to keep the acceptance rate high and improve mixing is to use auxiliary variable methods. Briefly, these rely on the notion that while sampling from $p(\boldsymbol{f}|\boldsymbol{y}_{1:N})$ may be difficult, we can introduce an auxiliary variable, $\boldsymbol{f}_c = \boldsymbol{f}(x_c)$, and sample from $p(\boldsymbol{f}, \boldsymbol{f}_c|\boldsymbol{y}_{1:N})$ by first sampling $p(\boldsymbol{f}|\boldsymbol{f}_c, \boldsymbol{y}_{1:N})$ and then sampling $p(\boldsymbol{f}_c|\boldsymbol{y}_{1:N})$. That is

$$p(\boldsymbol{f}|\boldsymbol{y}_{1:N}) = \int_{\boldsymbol{f}_c} d\boldsymbol{f}_c p(\boldsymbol{f}|\boldsymbol{f}_c, \boldsymbol{y}_{1:N}) p(\boldsymbol{f}_c|\boldsymbol{y}_{1:N}) \tag{7.38}$$

where the integral is over the entire range of $\boldsymbol{f}_c$. This, in principle, does not solve our problem as sampling from $p(\boldsymbol{f}|\boldsymbol{f}_c, \boldsymbol{y}_{1:N})$ may be just as difficult as sampling from $p(\boldsymbol{f}|\boldsymbol{y}_{1:N})$ in the first place. However, under the approximation that $\boldsymbol{f}_c$ is an approximate sufficient statistic for $\boldsymbol{f}$, then $p(\boldsymbol{f}|\boldsymbol{f}_c, \boldsymbol{y}_{1:N})$ can be approximated as $p(\boldsymbol{f}|\boldsymbol{f}_c)$.

Now, for this auxiliary variable scheme, the proposal distribution is

$$Q_{\boldsymbol{f}_c^{old}}(\boldsymbol{f}_c^{prop}, \boldsymbol{f}^{prop}) = p(\boldsymbol{f}^{prop}|\boldsymbol{f}_c^{prop}) q(\boldsymbol{f}_c^{prop}|\boldsymbol{f}_c^{old}). \tag{7.39}$$

For now, we write down our acceptance ratio as

$$R_{\boldsymbol{f}^{old}}(\boldsymbol{f}^{prop}) = \frac{p(\boldsymbol{y}_{1:N}|\boldsymbol{f}^{prop}, \boldsymbol{f}_c^{prop}) p(\boldsymbol{f}^{prop}, \boldsymbol{f}_c^{prop})}{p(\boldsymbol{y}_{1:N}|\boldsymbol{f}^{old}, \boldsymbol{f}_c^{old}) p(\boldsymbol{f}^{old}, \boldsymbol{f}_c^{old})} \frac{Q_{\boldsymbol{f}_c^{prop}}(\boldsymbol{f}_c^{old}, \boldsymbol{f}^{old})}{Q_{\boldsymbol{f}_c^{old}}(\boldsymbol{f}_c^{prop}, \boldsymbol{f}^{prop})} \tag{7.40}$$

$$= \frac{p(\boldsymbol{y}_{1:N}|\boldsymbol{f}^{prop}) p(\boldsymbol{f}^{prop}|\boldsymbol{f}_c^{prop}) p(\boldsymbol{f}_c^{prop})}{p(\boldsymbol{y}_{1:N}|\boldsymbol{f}^{old}) p(\boldsymbol{f}^{old}|\boldsymbol{f}_c^{old}) p(\boldsymbol{f}_c^{old})} \frac{Q_{\boldsymbol{f}_c^{prop}}(\boldsymbol{f}_c^{old}, \boldsymbol{f}^{old})}{Q_{\boldsymbol{f}_c^{old}}(\boldsymbol{f}_c^{prop}, \boldsymbol{f}^{prop})} \tag{7.41}$$

$$= \frac{p(\boldsymbol{y}_{1:N}|\boldsymbol{f}^{prop}) p(\boldsymbol{f}_c^{prop})}{p(\boldsymbol{y}_{1:N}|\boldsymbol{f}^{old}) p(\boldsymbol{f}_c^{old})} \frac{q(\boldsymbol{f}_c^{old}, \boldsymbol{f}^{old})}{q(\boldsymbol{f}_c^{prop}, \boldsymbol{f}^{prop})} \tag{7.42}$$

where, in going for the first to second line, we dropped the $\boldsymbol{f}_c^{prop}$ dependence of the likelihood and broke the prior down into two terms. We, again, are free to choose the prior for q as this will also satisfy the smoothness assumptions. Then we can find the location of the control points as well as their number in order to optimize the acceptance ratio.

Algorithm 7.1: Gaussian Process sampler with control points

1. Start at a feasible values $\boldsymbol{f}^{(0)}$, $\boldsymbol{f}_c^{(0)}$, L' and $\boldsymbol{x}_c$
2. For each j from 1 up to J repeat:
   - For each i from 1 up to L' repeat:
     - Sample $f_{c_i}^{prop} \sim p(f_{c_i}^{prop}|\boldsymbol{f}_{c_{-i}}^{(j)})$
     - Sample $\boldsymbol{f}^{prop} \sim p(\boldsymbol{f}^{prop}|f_{c_i}^{(j+1)}, \boldsymbol{f}_{c_{-i}}^{(j)})$
     - Accept or reject $(\boldsymbol{f}^{(j+1)}, f_{c_i}^{(j+1)})$ according to eq. (7.42)

Upon completion, the sampler produces $(\boldsymbol{f}^{(j)}, \boldsymbol{f}_c^{(j)})$ for $j = 1, \ldots, J$.

The goal in selecting the location of the control points is to minimize the variance of $p(\boldsymbol{f}^{prop}|\boldsymbol{f}_c^{prop})$ in order to maximize the probability of acceptance of the proposed $\boldsymbol{f}^{prop}$. In this case, the covariance of $p(\boldsymbol{f}^{prop}|\boldsymbol{f}_c^{prop})$, cov given in eq. (7.23), is an explicit function of the number, L', and location of the control points, $\boldsymbol{f}_c$,

$$cov(\boldsymbol{x}_c, L') = \mathsf{Tr}(\tilde{\boldsymbol{K}}) = \mathsf{Tr}(\boldsymbol{K}_{**} - \boldsymbol{K}_*^T \cdot (\boldsymbol{K})^{-1} \cdot \boldsymbol{K}_*) \tag{7.43}$$

where the $\boldsymbol{K}_{**}$ denotes the covariance between the locations of $\boldsymbol{f}$. To minimize $cov(\boldsymbol{x}_c, L')$, we may add a point and vary its location until $cov(\boldsymbol{x}_c, L')$ is minimized. Then keep adding points, subject to fixed previous points, and vary their location until their new location minimizes $cov(\boldsymbol{x}_c, L')$, then stop once $cov(\boldsymbol{x}_c, L')$ has fallen below a preset threshold (e.g., until the covariance of $p(\boldsymbol{f}|\boldsymbol{f}_c)$ falls below some fraction of the variance of $p(\boldsymbol{f})$).

7.1.9 Sampling over hyperparameters in the Gaussian Process prior

Within a Bayesian setting, with hyperparameters denoted by $\boldsymbol{\alpha}$ and model parameters $\boldsymbol{f}$, the objective is to construct $p(\boldsymbol{\alpha}, \boldsymbol{f}|\boldsymbol{y}_{1:N})$.

We imagine a two part Gibbs sampling scheme where we first sample $\boldsymbol{\alpha}$ from $p(\boldsymbol{\alpha}|\boldsymbol{f}, \boldsymbol{y}_{1:N})$ then $\boldsymbol{f}$ from $p(\boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{y}_{1:N})$.

At this point, computing $p(\boldsymbol{\alpha}|\boldsymbol{f}, \boldsymbol{y}_{1:N})$ requires that we specify a prior over hyperparameters, $p(\boldsymbol{\alpha})$. Depending on the likelihood, the conjugacy of the likelihood may not be guaranteed forcing a Metropolis-Hastings step for $p(\boldsymbol{\alpha}|\boldsymbol{f}, \boldsymbol{y}_{1:N})$ within Gibbs.

7.1.10 Classification with Gaussian Process prior

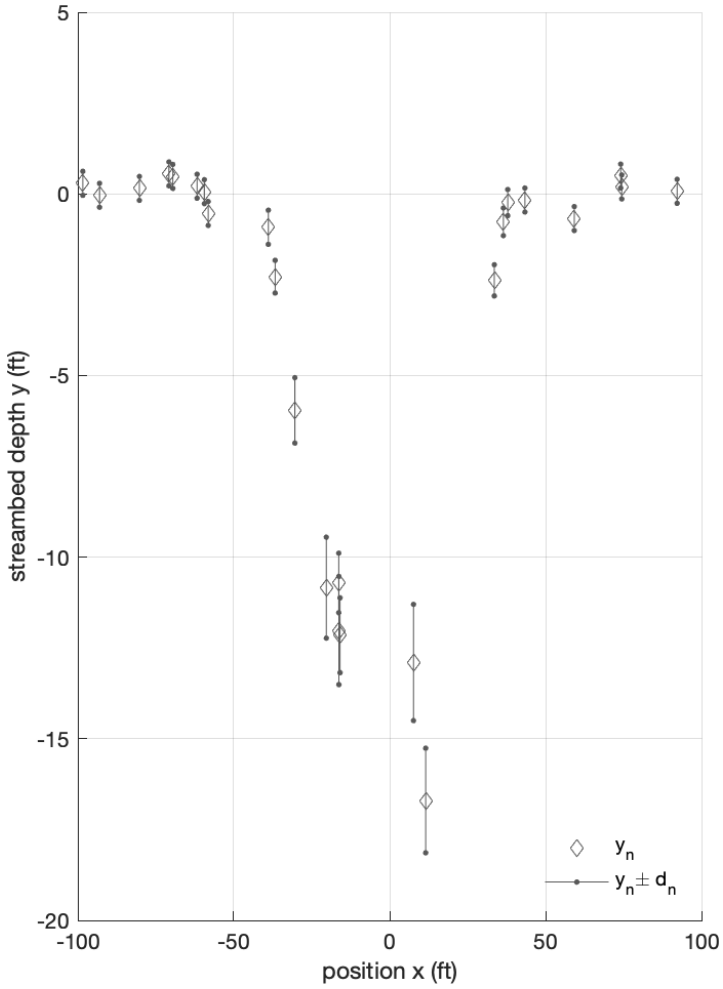
(Optional subsection)

7.2 Exercise problems

Exercise 7.1

Simulate a polynomial curve with Gaussian noise. Implement the Gaussian process to learn the shape of the underlying curve. Use a zero mean Gaussian process prior with a squared exponential covariance matrix. To present your results, plot the posterior mean curve estimate (in a thick line) and all other curves sampled from your posterior as thin lines.

Exercise 7.2: Stream-bed profile estimation



x_n (ft)	y_n (ft)	d_n (ft)
-16.33	-10.71	0.82
43.37	-0.17	0.33
-98.40	0.29	0.33
-38.91	-0.92	0.47
-69.54	0.48	0.33
-80.25	0.15	0.33
-61.76	0.21	0.33
-30.40	-5.96	0.90
-20.32	-10.84	1.39
7.64	-12.90	1.60
-15.91	-12.15	1.03
36.46	-0.77	0.38
-58.18	-0.54	0.33
74.43	0.19	0.33
-93.03	-0.04	0.33
33.56	-2.38	0.43
-16.28	-12.02	1.49
11.55	-16.70	1.44
-70.79	0.55	0.33
-59.43	0.06	0.33
59.20	-0.68	0.33
92.18	0.07	0.33
-36.73	-2.28	0.45
37.86	-0.24	0.36
74.09	0.49	0.33

The dataset shown above provides measurements of the depth, y_n, of a water channel obtained at randomly chosen positions, x_n, along its cross-section. Each depth measurement, y_n, is contaminated with additive normal noise of zero mean and standard deviation, d_n, which is shown by the error-bars.

1. Formulate a Bayesian regression model employing a Gaussian process to estimate the depth profile along the channel's cross-section. In the Gaussian processes prior make your own choices for the mean and co-variance and reason on your selection.
2. Use Monte Carlo sampling to characterize your prior depth profiles and summarize the results graphically.
3. Use Monte Carlo sampling to characterize your posterior depth profiles and summarize the results graphically.
4. Compute your MAP depth profile estimate and summarize the results graphically.
5. At each of the positions

$$x_A = -20, \qquad x_B = -10, \qquad x_C = 0, \qquad x_D = +10, \qquad x_E = +20 \text{ ft}$$

along the channel's cross-section, compute the posterior probability that the depth is less than 10 ft. Derive your results analytically and also using Monte Carlo sampling.

6. Compute the posterior probability that the depth is less than 10 ft in all positions x_A, x_B, x_C, x_D, x_E simultaneously. Derive your results analytically and also using Monte Carlo sampling.

Chapter 8

Hidden Markov models

By the end of this chapter, we will have presented
- *The fundamentals of hidden Markov models*
- *Specialized computational algorithms*
- *Various ways of modeling dynamics*

In this chapter we are exclusively concerned with modeling *time dependent measurements*. Specifically, we revisit some of the systems introduced in chapter 2 and present, in a unified framework, several methods to combine dynamic and observation likelihoods. It will become apparent soon that computational tractability is by no means guaranteed in time dependent problems and often we need to consider specialized algorithms that built upon or extend those of chapter 5. For this reason, we also present appropriate computational methods that can be used to train the resulting models in an efficient manner. Due to their simplicity, here we focus on modeling *discrete systems* evolving in *discrete time;* while, we present more general systems in the subsequent chapter.

8.1 Introduction

Throughout this chapter, we consider a system that has access to any of discretely many states similar to the systems we saw in section 2.4. For convenience, throughout this chapter, we denote the constitutive states of the system with σ_m, and use $m = 1, \dots, M$ to distinguish between them. The number of different states, M, that the system may occupy can be finite or even infinite and depends heavily upon the problem at hand.

When a system like this evolves through time, and so its particular position within its state-space may differ from one time point to another, the most fundamental questions that we may ask are *"what is the sequence of successive states that the system goes through over time?"* or *"what are the properties of each state that the system goes through over time?"*

To be able to formulate our questions more precisely, we consider ordered time levels t_n, which we index with $n = 1, \dots, N$ and we use s_n to denote the state occupied by the system at t_n. That is, for a given n, the occupying state s_n takes its value from the constituting states $\sigma_1, \dots, \sigma_M$. So, our questions about the system at hand can be answered by seeking to estimate the trajectory $s_{1:N}$.

Note 8.1: Label and index conventions

As the systems we met in note 2.5, only the labeled states σ_m have a modeling meaning; while, the labels m themselves are meaningless in our framework. Nevertheless, such distinction does not carry over the indices n of the time levels. By convention, our time levels are ordered $t_{n-1} < t_n$, which indicates that, contrary to the labels m, our indices n carry important modeling information themselves.

Of course, with no observations, it is pointless to seek any conclusion about the trajectory $s_{1:N}$. So, suppose that whenever the system occupies a state σ_m, it generates observations according to a probability distribution $\mathbb{F}_{\sigma_m}$, or its associated density $F_{\sigma_m}(w)$, that is unique to σ_m.

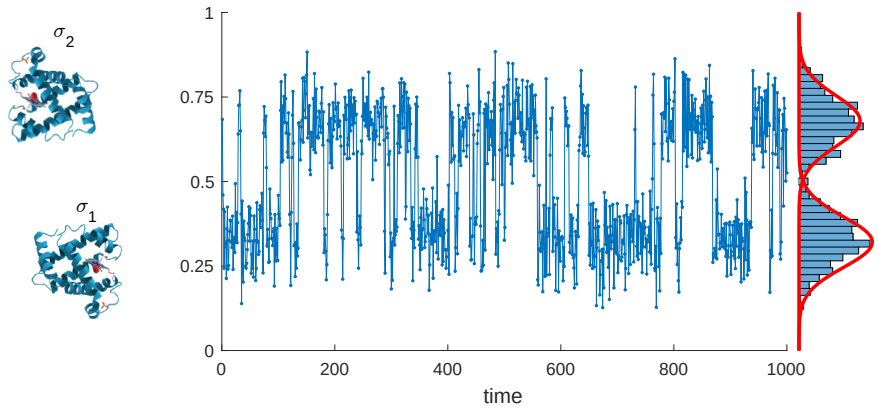


Figure 8.1: An example of a protein with 2 conformational states σ_1 and σ_2 and a hypothetical observed signal. In this example, observations from σ_1 are associated with signal values accumulated around 0.70, while observations from σ_2 are accumulated around 0.30. Due to excessive noise, observations are widely distributed around the central values, indicating that the corresponding emission distributions F_{σ_1} and F_{σ_2} resemble probability distribution with wide support.

Example 8.1

In this example, we consider a biomolecule, such as a protein, that can be in two conformations: one that we abstractly denote σ_1 and a different one that we denote σ_2, see fig. 8.1. Suppose that we observe the biomolecule for a time period and that we do so by monitoring a scalar quantity, for example FRET efficiency.

Observations generated with the biomolecule in σ_1 and σ_2 will be accumulated around different values, ϕ_{σ_1} and ϕ_{σ_2}. Due to unavoidable noise, our observations will be spread over a range, wide or narrow depending upon our physical setup, around ϕ_{σ_1} and ϕ_{σ_2}, see fig. 8.1.

Because of the "spreading" caused by noise, it is more natural to consider observation probability distributions, $\mathbb{F}_{\sigma_1}$ and $\mathbb{F}_{\sigma_2}$, associated with the two states rather than just the values ϕ_{σ_1} and ϕ_{σ_2}. The associated probability densities, $F_{\sigma_1}(w)$ and $F_{\sigma_2}(w)$ are shown in fig. 8.1.

To be able to derive a concise formulation, we assume that we make only *one* observation, which we denote with w_n, per time level t_n. In other words, our *assessment rules* are

$$w_n|s_n \sim \mathbb{F}_{s_n}, \qquad n = 1, \dots, N. \tag{8.1}$$

Of course, each time level's observation may consist of more than one scalar quantities, *i.e.* observations may be array-valued.

Equation (8.1) provides the foundation of our formulation. As time does not appear explicitly in this formulation, the particular order observations $w_n|s_n$ are made is irrelevant, so we may use the concepts of chapter 6 to answer our questions about the system at hand. However, for most physical systems, often the *order in which observations are made* provides important information that we wish to incorporate into our formulation. Namely, in most cases we want to incorporate dynamics in the sense that whenever the system passes from a particular state σ_m at a time level t_n its subsequent transitions and generated observations follow similar statistics.

Note 8.2: Why do we need dynamics?

With the introduction of dynamics into the assessment rule, eq. (8.1), we hope to obtain better estimates of the trajectory $s_{1:N}$ or of the particular characteristics of each $\mathbb{F}_{\sigma_m}$ than when we consider each s_n independent from the others. This is especially useful when $\mathbb{F}_{\sigma_m}$ are overlapping and so each measurement w_n alone does not suffice to determine conclusively each s_n.

As we have seen in chapter 2, dynamics for systems with discrete state-spaces evolving in discrete time are best described by assigning appropriate distributions on the occupying states s_1 and $s_n|s_{n-1}$ that we term the *initialization* and *transition rules*. Next, we discuss some modeling options to consider when choosing such distributions.

8.2 The Hidden Markov Model

8.2.1 Modeling dynamics

From the modeling point of view, the simplest and often the most convenient way to incorporate dynamics into an observation model like eq. (8.1) is to adopt transition probabilities between any pair σ_m and $\sigma_{m'}$ of states in the system's state-space. By "convenient" here we mean that, as discussed below, such formulations generally lead to intuitive and computationally tractable problems.

Generally, the system's transitions need not be reversible, so we may adopt different probabilities for transitions $\sigma_m \to \sigma_{m'}$ and $\sigma_{m'} \to \sigma_m$. In the most general case, we let $\pi_{\sigma_m \to \sigma_{m'}}$ denote the probability of the system starting at σ_m and, within *one* time step, changing to $\sigma_{m'}$. In this setup, none of $\pi_{\sigma_m \to \sigma_{m'}}$ can be negative; nevertheless, some $\pi_{\sigma_m \to \sigma_{m'}}$ can be zero indicating that the system cannot undergo transitions $\sigma_m \to \sigma_{m'}$ in a single step.

To facilitate the presentation that follows, we gather all transition probabilities out of the same state σ_m into an array $\boldsymbol{\pi}_{\sigma_m} = [\pi_{\sigma_m \to \sigma_1}, \pi_{\sigma_m \to \sigma_2}, \cdots, \pi_{\sigma_m \to \sigma_M}]$. Because once the system departs from σ_m it necessarily lands somewhere *within* the state-space $\sigma_{1:M}$, the individual transition probabilities assigned must satisfy $\sum_{\sigma_{m'}} \pi_{\sigma_m \to \sigma_{m'}} = 1$. Consequently, each $\boldsymbol{\pi}_{\sigma_m}$ is, in fact, a *probability* vector.

Note 8.3: Transition probability matrix

To simplify the notation, we tabulate the transition probabilities into

$$\begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_M \end{array} \begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \pi_{\sigma_1 \to \sigma_1} & \pi_{\sigma_1 \to \sigma_2} & \cdots & \pi_{\sigma_1 \to \sigma_M} \\ \pi_{\sigma_2 \to \sigma_1} & \pi_{\sigma_2 \to \sigma_2} & \cdots & \pi_{\sigma_2 \to \sigma_M} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{\sigma_M \to \sigma_1} & \pi_{\sigma_M \to \sigma_2} & \cdots & \pi_{\sigma_M \to \sigma_M} \end{bmatrix} \end{array} = \begin{bmatrix} \boldsymbol{\pi}_{\sigma_1} \\ \boldsymbol{\pi}_{\sigma_2} \\ \vdots \\ \boldsymbol{\pi}_{\sigma_M} \end{bmatrix} = \boldsymbol{\Pi}.$$

This matrix is similar to the transition probability matrices we encountered in chapter 2.

Under this formulation, dynamics are represented generically by the transition rules

$$s_n|s_{n-1} \sim \mathsf{Categorical}_{\sigma_{1:M}}\left(\boldsymbol{\pi}_{\sigma_m}\right), \qquad n = 2, \ldots, N. \tag{8.2}$$

The initial state of the system s_1 is not included in eq. (8.2) since there is no predecessor occupying state. So, we have to adopt separate probabilities, which we denote with $\boldsymbol{\rho} = [\rho_{\sigma_1}, \rho_{\sigma_2}, \cdots, \rho_{\sigma_M}]$. So, eq. (8.2) is combined with the initialization rule

$$s_1 \sim \mathsf{Categorical}_{\sigma_{1:M}}\left(\boldsymbol{\rho}\right), \tag{8.3}$$

which completes the description of the system's dynamics.

Note 8.4: Deterministic initialization

When the initial state of our dynamical system is specified deterministically, we may still maintain the same formulation by simply setting $\rho_{\sigma_m} = 1$ for the constituting state σ_m that the system starts in and $\rho_{\sigma_{m'}} = 0$ for every other one. For example, for a system that is initialized at σ_2, the initial probabilities are $\boldsymbol{\rho} = [0, 1, \cdots, 1]$.

8.2.2 Modeling observations

To proceed with the formulation it is easier to adopt state-specific parameters ϕ_{σ_m}, one for each constitutive σ_m, and express the emission distributions in a parametrized fashion $\mathbb{F}_{\sigma_m} = \mathbb{G}_{\phi_{\sigma_m}}$, where the mother distribution $\mathbb{G}_\phi$ attains a form that depends on a generic parameter ϕ. With this convention, states σ_m are associated with unique values ϕ_{σ_m} and eq. (8.1) is recast as

$$w_n|s_n \sim \mathbb{G}_{\phi_{s_n}}, \qquad n = 1, \dots, N. \tag{8.4}$$

To facilitate the notation, we typically gather the emission parameters of all states into an array $\boldsymbol{\phi} = [\phi_{\sigma_1}, \phi_{\sigma_2}, \cdots, \phi_{\sigma_M}]$.

Note 8.5: Emission models

Generally, the mother distribution $\mathbb{G}_\phi$, or the functional form of its density $G(w;\phi)$, is motivated by physical considerations, for example observations corrupted by Brownian noise are typically represented by Normal distributions, while observations corrupted by shot noise are typically represented by Poisson distributions. At the crudest level, such distinction can be made based on whether the measurements $w_n|s_n$ are corrupted by *additive* or *multiplicative* noise.

Nearly always the parameters ϕ, which may consist of more than one scalar components, have a direct physical interpretation. In particular, in problems where ϕ is identified with the mean of $\mathbb{G}_\phi$, the individual emission parameters ϕ_{σ_m} are often termed *emission* or *state levels*.

Example 8.2: Emission distributions for FRET measurements

In fluorescent experiments relying on FRET measurements, typically, at each time level t_n, two scalar measurements are obtained, w_n^D and w_n^A. These are the number of photons emitted by a fluorophore designated as *donor* and the number of photons emitted by a second fluorophore designated as *acceptor,* respectively.

Because individual photons are emitted by the fluorophores independently, the raw measurements are described by

$$w_n^D|s_n \sim \text{Poisson}\left(\mu_{s_n}^D\right), \qquad w_n^A|s_n \sim \text{Poisson}\left(\mu_{s_n}^A\right), \qquad n = 1, \dots, N,$$

where s_n is the conformational state of the molecule attached to the two fluorophores and the state dependent parameters $\mu_{\sigma_1}^D, \dots, \mu_{\sigma_M}^D$ and $\mu_{\sigma_1}^A, \dots, \mu_{\sigma_M}^A$ are the corresponding photon accumulation levels.

Most often w_n^D and w_n^A are combined into a single scalar quantity

$$\epsilon_n = \frac{w_n^A}{w_n^A + w_n^D}$$

which is termed (apparent) FRET efficiency. In this case, the observation model takes a simpler form

$$\epsilon_n|s_n \sim \mathbb{G}_{\boldsymbol{\phi}_{\sigma_m}}$$

where $\boldsymbol{\phi}_{\sigma_m} = (\mu_{\sigma_m}^D, \mu_{\sigma_m}^A)$. In general, the probability density $G(\epsilon;\boldsymbol{\phi})$ cannot be derived analytically. However, provided *all* emission levels are high enough, such that we can safely use the approximations

$$\text{Poisson}\left(w^D;\mu^D\right) \approx \text{Gamma}\left(w^D;\mu^D,1\right), \qquad \text{Poisson}\left(w^A;\mu^A\right) \approx \text{Gamma}\left(w^A;\mu^A,1\right),$$

then $G(\epsilon;\boldsymbol{\phi})$ is well approximated by a Beta density

$$G\left(\epsilon;\mu^D,\mu^A\right) \approx \text{Beta}\left(\epsilon;\mu^A,\mu^D\right).$$

8.2.3 Modeling overview

In summary, the model described so far is

$$s_1|\boldsymbol{\rho} \sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\rho}) \tag{8.5}$$

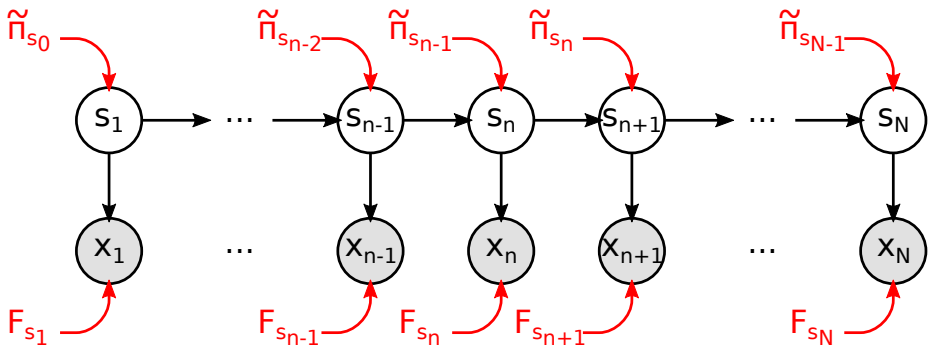


Figure 8.2: Graphical representation of a hidden Markov model. Here, observations x_n are dark shaded and parameters $\tilde{\pi}_{\sigma_m}$ and ϕ_{σ_m} are assumed known.

$$s_n|s_{n-1}, \boldsymbol{\Pi} \sim \text{Categorical}_{\sigma_{1:M}}\left(\boldsymbol{\pi}_{\sigma_m}\right), \qquad n = 2, \dots, N \tag{8.6}$$

$$w_n|s_n, \boldsymbol{\phi} \sim \mathbb{G}_{\phi_{s_n}}, \qquad n = 1, \dots, N \tag{8.7}$$

where, for clarity, we emphasize the dependencies upon the parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ by conditioning explicitly upon them. The three equations model: initialization, transitions, and observations of the system under study, respectively, and combined with a clear specification of the state-space $\sigma_{1:M}$, they provide a complete description of our problem.

This model is best known as the *hidden Markov model* (HMM). It contains two sets of parameters: dynamic, $\boldsymbol{\rho}$ and $\boldsymbol{\Pi}$, and observation $\boldsymbol{\phi}$. From the inference point of view, the trajectory $s_{1:N}$ gathers the occupying states, which are latent variables, and $w_{1:N}$ gathers the measurements. The dependences among the variables are illustrated in fig. 8.2.

The formulation of the HMM in eqs. (8.5) to (8.7) is very general and for this reason is one of the mostly used models for time series analysis. Since it is already in generative form, when tackling a direct problem, it is straightforward to use this model for the simulation of synthetic measurements $w_{1:N}$, for example through ancestral sampling, algorithm 1.3. Nevertheless, mostly we are interested in using this formulation to solve inverse problems.

In particular, with a HMM we are mostly interested in answering one or more of the following questions:

1. Given observations $w_{1:N}$ and parameter values $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ what is the likelihood of $w_{1:N}$?
2. Given observations $w_{1:N}$ and parameter values $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ what are the occupying states $s_{1:N}$?
3. Given observations $w_{1:N}$ what are the values of the parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$?

These questions are commonly referred to as: *evaluation, decoding,* and *estimation,* respectively. As we will see shortly, to answer them we can follow two complementary routes: a frequentist and a Bayesian. We describe these separately in the subsequent sections.

Note 8.6: Time indexing and missing observations

With the indexing convention we adopt here, we designate with $n = 1$ the *earliest* time level that has an observation and with $n = N$ the *latest* one. Further, we assume that *every* intermediate time level $n = 2, \dots, N-1$ has an observation too. This is a measurement-centric convention in the sense that the timing schedule of the measurement acquisition protocol determines the precise structure of the hidden state sequence.

Occasionally we might encounter situations that we need to incorporate time levels *without* observations, for example when modeling measurements collected at *irregular times*. In such situations, we may generalize our formulation in at least two possible ways:

1. Use the same indexing convention with precisely one observation per time level and adopt *time dependent kinetics*, for example by explicitly requiring transition probabilities $\pi_{n,\sigma_m \to \sigma_{m'}}$ that may change between time levels.

2. Use an indexing scheme with *redundant time levels*. In particular, we may chose to maintain a hidden state sequence at a finer, but regular, time spacing and associate only some of the occupying states with the observations while leaving the others unassociated with.

For both cases, the theory we present can be modified easily. In particular, if a HMM contains time levels that have no observations, either in the beginning, during, or at the end, such time levels can be modeled as having missing observations and the following theory can be readily extended to handle them.

8.3 The Hidden Markov Model in the frequentist paradigm

The concepts of this section are direct extensions of the material of the previous chapters, *e.g.* chapter 2. As we have already introduced them, here we treat mostly computational aspects specifically tailored for HMM. We will be re-using most of these algorithms also in the Bayesian context of the later sections. To increase clarity, we omit lengthy derivations of key equations in this section. We mark these equations with ** and list their derivation in appendix E.

8.3.1 Evaluation of the likelihood

Evaluation of a HMM asks for the computation of the (marginal) likelihood

$$p\left(w_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)=\sum_{s_{1:N}}p\left(w_{1:N},s_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)$$

with the sum of the completed likelihood, $p\left(w_{1:N},s_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)$, taken over every possible state sequence $s_{1:N}$.

Note 8.7: Computational complexity

Naive evaluation of this enormous sum, where a term

$$p\left(w_{1:N},s_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)=p\left(w_{1:N}|s_{1:N},\boldsymbol{\phi}\right)p\left(s_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi}\right)$$

is computed for each possible $s_{1:N}$ and added in a greedy manner, requires evaluation and addition of approximately M^N terms. This is prohibitively large even for small problems. Instead, below we describe a particular computational scheme, which is termed *filtering*, that scales as M^2N. Shortly, we will develop similar filtering schemes to answer also the other questions of the HMM.

This significant increase in the efficiency of the computations is possible on account of certain recursions permitted by the 1st order Markov property that links the occupying states across time in eq. (8.6). Unfortunately, such optimal scaling does not carry over higher order Markov systems which, in practice, severely limits our flexibility to utilize higher than 1st order dynamics.

Instead of completing with the entire state sequence $s_{1:N}$, the computation of the likelihood is achieved most efficiently by completing first only with respect to the terminal occupying state s_N, for instance, as follows

$$p\left(w_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)=\sum_{s_N}p\left(w_{1:N},s_N|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)=\sum_{s_N}\mathcal{A}_N\left(s_N\right). \tag{8.8}$$

This sum is readily computed as long as $\mathcal{A}_N(s_N)$ is available for all possible values of s_N which, written explicitly, are $\mathcal{A}_N(\sigma_1),\mathcal{A}_N(\sigma_2),\ldots,\mathcal{A}_N(\sigma_M)$. Such terms are defined, more generally, for any time level by the joint distributions

$$\mathcal{A}_n(s_n)=p(w_{1:n},s_n|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}),\qquad\qquad n=1,\ldots,N \tag{8.9}$$

and we may compute them recursively. In particular, our recursion relies on

$$\mathcal{A}_n(s_n)=G\left(w_n;\phi_{s_n}\right)\sum_{s_{n-1}}\pi_{s_{n-1}\to s_n}\mathcal{A}_{n-1}(s_{n-1}),\qquad\qquad n=2,\ldots,N \tag{*8.10*}$$

and requires the initial condition $\mathcal{A}_1(s_1)$ to iterate forward. As a direct consequence of the definition of $\mathcal{A}_1(s_1)$, we obtain the initial condition by

$$\mathcal{A}_1(s_1) = G(w_1; \phi_{s_1})\, \rho_{s_1}. \tag{*8.11*}$$

The steps involved are summarized in algorithm 8.1.

Algorithm 8.1: Forward recursion for HMM (unstable version)

Given observations $w_{1:N}$ and parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$, the forward terms $\mathcal{A}_n(\sigma_m)$, for each time level n and each state σ_m, are computed as follows:

- At $n = 1$ initialize by

$$\mathcal{A}_1(\sigma_m) = G(w_1; \phi_{\sigma_m})\rho_{\sigma_m}, \qquad m = 1, \ldots, M$$

- For $n = 2, \ldots, N$ compute recursively

$$\mathcal{A}_n(\sigma_m) = G(w_n; \phi_{\sigma_m}) \sum_{\sigma_{m'}} \pi_{\sigma_{m'} \to \sigma_m} \mathcal{A}_{n-1}(\sigma_{m'}), \qquad m = 1, \ldots, M$$

Upon completion, the algorithm provides every $\mathcal{A}_n(\sigma_m)$ which may be tabulated as

$$\begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_n \\ \vdots \\ t_N \end{array} \begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \mathcal{A}_1(\sigma_1) & \mathcal{A}_1(\sigma_2) & \cdots & \mathcal{A}_1(\sigma_M) \\ \mathcal{A}_2(\sigma_1) & \mathcal{A}_2(\sigma_2) & \cdots & \mathcal{A}_2(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}_n(\sigma_1) & \mathcal{A}_n(\sigma_2) & \cdots & \mathcal{A}_n(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}_N(\sigma_1) & \mathcal{A}_N(\sigma_2) & \cdots & \mathcal{A}_N(\sigma_M) \end{bmatrix} \end{array} \begin{array}{c} \\ \mathcal{A}_1(s_1) \\ \mathcal{A}_2(s_2) \\ \vdots \\ \mathcal{A}_n(s_n) \\ \vdots \\ \mathcal{A}_N(s_N) \end{array}$$

Note 8.8: Vectorization

Gathering the forward terms of the same time level in raw arrays

$$\mathbb{A}_n = \begin{bmatrix} \mathcal{A}_n(\sigma_1) & \mathcal{A}_n(\sigma_2) & \cdots & \mathcal{A}_n(\sigma_M) \end{bmatrix}, \qquad n = 1, \ldots, N$$

and similarly the likelihood terms also in row arrays

$$\mathbb{\Gamma}_n = \begin{bmatrix} G(w_n; \phi_{\sigma_1}) & G(w_n; \phi_{\sigma_2}) & \cdots & G(w_n; \phi_{\sigma_M}) \end{bmatrix}, \qquad n = 1, \ldots, N$$

the filtering recursions can be executed in vectorized form

$$\begin{aligned} \mathbb{A}_1 &= \mathbb{\Gamma}_n \odot \boldsymbol{\rho} \\ \mathbb{A}_n &= \mathbb{\Gamma}_n \odot (\mathbb{A}_{n-1}\boldsymbol{\Pi}), \qquad n = 2, \ldots, N \end{aligned}$$

where $\odot$ denotes the Hadamard (element-wise) product. If, instead of a row array, we represent $\mathbb{\Gamma}_n$ as a diagonal matrix $\mathbb{D}_{\mathbb{\Gamma}_n}$, then the filtering recursions take a more conventional form

$$\begin{aligned} \mathbb{A}_1 &= \boldsymbol{\rho}\, \mathbb{D}_{\mathbb{\Gamma}_n} \\ \mathbb{A}_n &= (\mathbb{A}_{n-1}\boldsymbol{\Pi})\, \mathbb{D}_{\mathbb{\Gamma}_n}, \qquad n = 2, \ldots, N \end{aligned}$$

that use only ordinary matrix-vector operations. From these two sets of filtering equations, the first one is preferable for computational implementations; while, the second one is often more convenient in theoretical derivations.

8.3.2 Decoding of the state sequence

Decoding of a HMM may be answered in at least two meaningful ways. Depending on the specifics, we might be interested in finding a single occupying state s_n^* that maximizes its respective (marginal) likelihood $p(s_n|w_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$; or in finding the sequence $s_{1:N}^\sharp$ that maximizes the entire (joint) likelihood $p(s_{1:N}|w_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$. Generally, individual states s_n^* are useful in problems where the optimal occupying state of a *particular time level* is sought after. In contrast, $s_{1:N}^*$ is useful in problems where the optimal trajectory over the *entire time course* is sought after.

Note 8.9

Collecting the occupying states s_n^* for all time levels n, we may also form a state sequence $s_{1:N}^*$. This sequence, however, must be used with caution because it might violate the kinetics in $\boldsymbol{\Pi}$. In particular, since $s_{1:N}^*$ considers each s_n^* *irrespective* of s_{n-1}^*; it may very well contain transitions $s_{n-1}^* \to s_n^*$ that correspond to probabilities $\pi_{s_{n-1}^* \to s_n^*} = 0$.

In contrast, $s_{1:N}^\sharp$ is *guaranteed* to obey the kinetics in $\boldsymbol{\Pi}$ because any sequence containing prohibited transitions $\pi_{s_{n-1}^\sharp \to s_n^\sharp}$ is already excluded.

Below, we see that the computation of each s_n^* requires a forward and a backward recursive passes. Similarly, the evaluation of $s_{1:N}^\sharp$ also requires two recursive passes; however, unlike s_n^*, where both passes need to utilize the observations $w_{1:N}$, the computation of $s_{1:N}^\sharp$ utilizes $w_{1:N}$ only in the forward pass.

Marginal decoding

To find each individual s_n^*, we need to compute $p(s_n|w_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$ for each n. This can be achieved efficiently, using the terms $\mathcal{A}_n(s_n)$ already mentioned. For instance, at the terminal $n = N$ time level

$$p(s_N|w_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \propto \mathcal{A}_N(s_N) \tag{*8.12*}$$

and for the earlier $n = 1, \ldots, N-1$ time levels

$$p(s_n|w_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \propto \mathcal{A}_n(s_n)\mathcal{B}_n(s_n). \tag{*8.13*}$$

Given $\mathcal{A}_n(s_n)$ and $\mathcal{B}_n(s_n)$, the sequence $s_{1:N}^*$ is readily computed by

$$s_n^* = \operatorname*{argmax}_{\sigma_m} \mathcal{A}_n(\sigma_m)\mathcal{B}_n(\sigma_m), \qquad n = 1, \ldots, N$$

which results directly from eqs. (*8.12*) and (*8.13*). The terms $\mathcal{B}_n(s_n)$ are defined by the distributions

$$\mathcal{B}_n(s_n) = p\left(w_{n+1:N}|s_n, \boldsymbol{\Pi}, \boldsymbol{\phi}\right), \qquad n = 1, \ldots, N-1 \tag{8.14}$$

and, similarly to $\mathcal{A}_n(s_n)$, may be also computed recursively. In particular, the recursion relies on

$$\mathcal{B}_n(s_n) = \sum_{s_{n+1}} \mathcal{B}_{n+1}(s_{n+1}) G\left(w_{n+1}; \phi_{s_{n+1}}\right) \pi_{s_n \to s_{n+1}}, \qquad n = 1, \ldots, N-1 \tag{*8.15*}$$

and requires the final condition $\mathcal{B}_N(s_N)$ to iterate backward. In view of eq. (*8.12*), we obtain the terminal condition, conventionally, by setting

$$\mathcal{B}_N(s_N) = 1. \tag{8.16}$$

The steps involved are summarized in algorithm 8.2.

Algorithm 8.2: Backward recursion for HMM (unstable version)

Given observations $w_{1:N}$ and parameters $\mathbf{\Pi}, \boldsymbol{\phi}$ the backward terms $\mathcal{B}_n(\sigma_m)$, for each time level n and each state σ_m, are computed as follows:

- At $n = N$ initialize by

$$\mathcal{B}_N(\sigma_m) = 1, \qquad m = 1, \dots, M$$

- For $n = N-1, \dots, 1$ compute recursively

$$\mathcal{B}_n(\sigma_m) = \sum_{\sigma_{m'}} \mathcal{B}_{n+1}(\sigma_{m'}) G\left(w_{n+1}; \phi_{\sigma_{m'}}\right) \pi_{\sigma_m \to \sigma_{m'}}, \qquad m = 1, \dots, M$$

Upon completion, the algorithm provides every $\mathcal{B}_n(\sigma_m)$ which may be tabulated as

$$\begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_n \\ \vdots \\ t_N \end{array} \begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \mathcal{B}_1(\sigma_1) & \mathcal{B}_1(\sigma_2) & \cdots & \mathcal{B}_1(\sigma_M) \\ \mathcal{B}_2(\sigma_1) & \mathcal{B}_2(\sigma_2) & \cdots & \mathcal{B}_2(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{B}_n(\sigma_1) & \mathcal{B}_n(\sigma_2) & \cdots & \mathcal{B}_n(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{B}_N(\sigma_1) & \mathcal{B}_N(\sigma_2) & \cdots & \mathcal{B}_N(\sigma_M) \end{bmatrix} \end{array} \begin{array}{c} \\ \mathcal{B}_1(s_1) \\ \mathcal{B}_2(s_2) \\ \vdots \\ \mathcal{B}_n(s_n) \\ \vdots \\ \mathcal{B}_N(s_N) \end{array}$$

Joint decoding

The computation of $s^\sharp_{1:N}$ relies on the factorization

$$p(s_{1:N}|w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}) = p(s_N|w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}) \prod_{n=1}^{N-1} p(s_n|s_{n+1:N}, w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi})$$

which implies that the maximizer $s^\sharp_{1:N}$ can be computed by the maximizers $s^\sharp_n$ of the individual factors $p(s_N|w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi})$ and $p(s_n|s^\sharp_{n+1:N}, w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi})$. In turn, given $\mathcal{A}_n(s_n)$, maximization of each factor can be simplified using

$$p(s_N|w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}) \propto \mathcal{A}_N(s_N) \qquad \text{(*8.17*)}$$

$$p(s_n|s_{n+1:N}, w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}) \propto \mathcal{A}_n(s_n)\pi_{s_n \to s_{n+1}}, \qquad n = 1, \dots, N-1. \qquad \text{(*8.18*)}$$

The steps involved, known as *Viterbi recursion*, are summarized in algorithm 8.3.

Algorithm 8.3: Viterbi recursion for discrete HMM

Given observations $w_{1:N}$, kinetic parameters $\mathbf{\Pi}$, and every $\mathcal{A}_n(\sigma_m)$, the Viterbi sequence $s^\sharp_{1:N}$ is computed as follows:
1) At $n = N$ initialize by

$$s^\sharp_N = \underset{\sigma_m}{\operatorname{argmax}}\, \mathcal{A}_N(\sigma_m)$$

2) For $n = N-1, \ldots, 1$ compute recursively

$$s^\sharp_n = \underset{\sigma_m}{\operatorname{argmax}}\, \mathcal{A}_n(\sigma_m)\pi_{\sigma_m \to s^\sharp_{n+1}}$$

8.3.3 Estimation of the parameters

Estimation of a HMM seeks the maximizer

$$(\boldsymbol{\rho}^*, \mathbf{\Pi}^*, \boldsymbol{\phi}^*) = \underset{\boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}}{\operatorname{argmax}}\, p\left(w_{1:N} | \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}\right).$$

of the likelihood. Completing the likelihood $p\left(w_{1:N} | \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}\right)$ with the state sequence $s_{1:N}$, for instance, as

$$p\left(w_{1:N} | \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}\right) = \sum_{s_{1:N}} p\left(s_{1:N}, w_{1:N} | \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}\right)$$

we may perform this maximization with an EM procedure, where we iterate between an Expectation (E) step and a Maximization (M) step, similar to section 3.5. The entire procedure applied on the HMM is known as Baum-Welch method and the steps involved are summarized in algorithm 8.4. In the next sections, we describe the steps of algorithm 8.4 in detail.

Algorithm 8.4: Baum-Welch algorithm

Given observations $w_{1:N}$ and an initial guess for the model parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ the Baum-Welch method computes successively improved approximations of the maximizer of $p\left(w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)$ by repeating the steps

• E-step:

—Use $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ to compute $\mathcal{A}_n(\sigma_m)$ and $\mathcal{B}_n(\sigma_m)$

—Use $\mathcal{A}_n(\sigma_m)$ and $\mathcal{B}_n(\sigma_m)$ to compute

$$\zeta_n(\sigma_m) = \mathcal{A}_n(\sigma_m)\mathcal{B}_n(\sigma_m)$$
$$\eta_n(\sigma_m, \sigma_{m'}) = \mathcal{A}_{n-1}(\sigma_m)\mathcal{B}_n(\sigma_{m'})G\left(w_n; \phi_{\sigma_{m'}}\right)\pi_{\sigma_m \to \sigma_{m'}}$$

—Use $\eta_n(\sigma_m, \sigma_{m'})$ to compute

$$\xi_{\sigma_m}(\sigma_{m'}) = \sum_{n=2}^{N} \eta_n(\sigma_m, \sigma_{m'})$$

• M-step:

—Update $\boldsymbol{\rho}$ by replacing with

$$\left(\frac{\zeta_1(\sigma_1)}{\sum_{\sigma_m}\zeta_1(\sigma_m)}, \cdots, \frac{\zeta_1(\sigma_M)}{\sum_{\sigma_m}\zeta_1(\sigma_m)}\right)$$

—Update $\boldsymbol{\Pi}$ by replacing each $\boldsymbol{\pi}_{\sigma_m}$ with

$$\left(\frac{\xi_{\sigma_m}(\sigma_1)}{\sum_{\sigma_{m'}}\xi_{\sigma_m}(\sigma_{m'})}, \cdots, \frac{\xi_{\sigma_m}(\sigma_M)}{\sum_{\sigma_{m'}}\xi_{\sigma_m}(\sigma_{m'})}\right)$$

—Update $\boldsymbol{\phi}$ by replacing each ϕ_{σ_m} with the maximizer of

$$\sum_{n=1}^{N} \zeta_n(\sigma_m) \log G\left(w_n; \phi_{\sigma_m}\right)$$

The iterations are terminated either after a fixed number of repetitions, or when the improvement between successive approximations of $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ falls below a certain threshold.

Note 8.10: Diagnosing convergence

Unfortunately, like any EM method, convergence of algorithm 8.4 to the *global* optimizer $\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*$ of $p\left(w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)$ is *not* guaranteed. In practice, we need to repeat the same procedure with multiple starting points that span a wide region of the entire parameter space. Such a strategy is computationally expensive; however, comparing the resulting optimizers $\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*$, we can diagnose sub-optimal ones caused by trapping in regions of local maxima. Because the algorithm 8.4 does not provide the value of $p\left(w_{1:N}|\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*\right)$, to compare the resulting maximizers, we need to compute it with another method. For example, through eq. (8.8) as we describe in section 8.3.1.

Expectation step*

In the E-step, we start from an initial approximation $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ of the maximizer $\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*$ we are after and prepare the expectation function to be maximized in the M-step. Specifically, we compute the expectation of

$$\log p\left(s_{1:N}, w_{1:N} | \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) = \log \rho_{s_1} + \sum_{n=2}^{N} \log \pi_{s_{n-1} \to s_n} + \sum_{n=1}^{N} \log G\left(w_n; \phi_{s_n}\right) \tag{*8.19*}$$

with respect to the probability distribution of $s_{1:N} | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$. Since this expectation is a function of $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$; while, it also depends upon $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$, we denote it with $Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$. In particular, this expectation is given by

$$\begin{aligned} Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) = & \sum_{s_1} p\left(s_1 | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right) \log \rho_{s_1} \\ & + \sum_{s_{n-1}} \sum_{n=2}^{N} \sum_{s_n} p\left(s_{n-1}, s_n | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right) \log \pi_{s_{n-1} \to s_n} \\ & + \sum_{s_n} \sum_{n=1}^{N} p\left(s_n | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right) \log G\left(w_n; \phi_{s_n}\right). \end{aligned} \tag{*8.20*}$$

The distributions of $s_n | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ and $s_{n-1}, s_n | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ can be computed in terms of the forward and backward terms of eqs. (8.9) and (8.14). Because these are computed based on $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$, we denote them with $\mathcal{A}_n^{\text{old}}(s_n), \mathcal{B}_n^{\text{old}}(s_n)$. In particular, the distributions are

$$p\left(s_n | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right) = \frac{\mathcal{A}_n^{\text{old}}(s_n) \mathcal{B}_n^{\text{old}}(s_n)}{p\left(w_{1:N} | \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right)} \tag{*8.21*}$$

$$p\left(s_{n-1}, s_n | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right) = \frac{\mathcal{A}_{n-1}^{\text{old}}(s_{n-1}) \mathcal{B}_n^{\text{old}}(s_n) G\left(w_n; \phi_{s_n}^{\text{old}}\right) \pi_{s_{n-1} \to s_n}^{\text{old}}}{p\left(w_{1:N} | \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right)}. \tag{*8.22*}$$

Finally, because $p\left(w_{1:N} | \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right)$ does not depend upon $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$, this term does not affect the maximization of $Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$. So, we can safely drop it to obtain

$$\begin{aligned} Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \propto & \sum_{\sigma_m} \zeta_1^{\text{old}}(\sigma_m) \log \rho_{\sigma_m} \\ & + \sum_{\sigma_m} \sum_{n=2}^{N} \sum_{\sigma_{m'}} \eta_n^{\text{old}}(\sigma_m, \sigma_{m'}) \log \pi_{\sigma_m \to \sigma_{m'}} \\ & + \sum_{\sigma_m} \sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \log G\left(w_n; \phi_{\sigma_m}\right) \end{aligned} \tag{8.23}$$

where we use the following definitions

$$\zeta_n^{old}(\sigma_m) = \mathcal{A}_n^{old}(\sigma_m) \mathcal{B}_n^{old}(\sigma_m), \qquad n = 1, \ldots, N \tag{8.24}$$

$$\eta_n^{old}(\sigma_m, \sigma_{m'}) = \mathcal{A}_{n-1}^{old}(\sigma_m) \mathcal{B}_n^{old}(\sigma_{m'}) G\left(w_n; \phi_{\sigma_{m'}}^{old}\right) \pi_{\sigma_m \to \sigma_{m'}}^{old}, \qquad n = 2, \ldots, N \tag{8.25}$$

*This is an advanced topic and could be skipped on a first reading.

Maximization step*

In the M-step, we obtain an improved approximation $\boldsymbol{\rho}^{\text{new}}, \boldsymbol{\Pi}^{\text{new}}, \boldsymbol{\phi}^{\text{new}}$ of the maximizer $\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*$ we are after by maximizing the expectation function prepared in the E-step. Specifically, we maximize $Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$ under the constraints

$$\begin{aligned} \sum_{\sigma_m} \rho_{\sigma_m} &= 1 \\ \sum_{\sigma_{m'}} \pi_{\sigma_m \to \sigma'_m} &= 1, \qquad\qquad m = 1, \ldots, M \end{aligned}$$

which are needed to ensure that $\boldsymbol{\rho}^{\text{new}}, \boldsymbol{\Pi}^{\text{new}}$ consist of valid probability vectors. Because our objective in eq. (8.23) is separable, the computation of the new maximizer can be broken into separate maximizations:

$$\begin{aligned} \boldsymbol{\rho}^{\text{new}} &= \underset{\boldsymbol{\rho}}{\operatorname{argmax}} \sum_{\sigma_m} \zeta_1^{\text{old}}(\sigma_m) \log \rho_{\sigma_m} \\ \boldsymbol{\pi}_{\sigma_m}^{\text{new}} &= \underset{\boldsymbol{\pi}_{\sigma_m}}{\operatorname{argmax}} \sum_{n=2}^{N} \sum_{\sigma_{m'}} \eta_n^{\text{old}}(\sigma_m, \sigma_{m'}) \log \pi_{\sigma_m \to \sigma_{m'}}, \qquad m = 1, \ldots, M \\ \phi_{\sigma_m}^{\text{new}} &= \underset{\phi_{\sigma_m}}{\operatorname{argmax}} \sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \log G\left(w_n; \phi_{\sigma_m}\right), \qquad m = 1, \ldots, M \end{aligned}$$

with each one held under the appropriate constraint.

Maximization for initial probabilities The first optimization entails one constraint, so we can solve it by using a single multiplier λ. Specifically, the corresponding Lagrangian is

$$\mathcal{L}\left(\lambda, \rho_{\sigma_1}, \ldots, \rho_{\sigma_M}\right) = \left(1 - \sum_{\sigma_m} \rho_{\sigma_m}\right) \lambda + \sum_{\sigma_m} \zeta_1^{\text{old}}(\sigma_m) \log \rho_{\sigma_m}$$

Accordingly, the optimizer solves

$$\begin{aligned} \frac{\partial \mathcal{L}\left(\lambda, \rho_{\sigma_1}, \ldots, \rho_{\sigma_M}\right)}{\partial \lambda} &= 0 \\ \frac{\partial \mathcal{L}\left(\lambda, \rho_{\sigma_1}, \ldots, \rho_{\sigma_M}\right)}{\partial \rho_{\sigma_m}} &= 0, \qquad\qquad m = 1, \ldots, M \end{aligned}$$

This system can be solved analytically. Specifically, the solution, which provides the improved value $\boldsymbol{\rho}^{\text{new}}$ of the optimizer $\boldsymbol{\rho}^*$ we are after, is

$$\boldsymbol{\rho}^{\text{new}} = \left(\frac{\zeta_1^{\text{old}}(\sigma_1)}{\sum_{\sigma_m} \zeta_1^{\text{old}}(\sigma_m)}, \cdots, \frac{\zeta_1^{\text{old}}(\sigma_M)}{\sum_{\sigma_m} \zeta_1^{\text{old}}(\sigma_m)} \right) \qquad \text{(*8.26*)}$$

Maximization for transition probabilities For each m, the second optimization entails also one constraint, so we can solve it by using a single multiplier κ_m, too. Specifically, the corresponding Lagrangian is

$$\mathcal{K}_m\left(\kappa_m, \pi_{\sigma_m \to \sigma_1}, \ldots, \pi_{\sigma_m \to \sigma_M}\right) = \left(1 - \sum_{\sigma_{m'}} \pi_{\sigma_m \to \sigma_{m'}}\right) \kappa_m + \sum_{n=2}^{N} \sum_{\sigma_{m'}} \eta_n^{\text{old}}(\sigma_m, \sigma_{m'}) \log \pi_{\sigma_m \to \sigma_{m'}}$$

*This is an advanced topic and could be skipped on a first reading.

Accordingly, the optimizer solves

$$\frac{\partial \mathcal{K}_m\left(\kappa_m, \pi_{\sigma_m \to \sigma_1}, \ldots, \pi_{\sigma_m \to \sigma_M}\right)}{\partial \kappa_m} = 0$$
$$\frac{\partial \mathcal{K}_m\left(\kappa_m, \pi_{\sigma_m \to \sigma_1}, \ldots, \pi_{\sigma_m \to \sigma_M}\right)}{\partial \pi_{\sigma_m \to \sigma_{m'}}} = 0, \qquad m' = 1, \ldots, M$$

Again, this system can be solved analytically. Specifically, the solution, which provides the improved value $\boldsymbol{\pi}_{\sigma_m}^{\text{new}}$ of the optimizer $\boldsymbol{\pi}_{\sigma_m}^{*}$ we are after, is

$$\boldsymbol{\pi}_{\sigma_m}^{\text{new}} = \left(\frac{\xi_{\sigma_m}^{\text{old}}(\sigma_1)}{\sum_{\sigma_{m'}} \xi_{\sigma_m}^{\text{old}}(\sigma_{m'})}, \cdots, \frac{\xi_{\sigma_m}^{\text{old}}(\sigma_M)}{\sum_{\sigma_{m'}} \xi_{\sigma_m}^{\text{old}}(\sigma_{m'})} \right) \tag{*8.27*}$$

where we use the following definition

$$\xi_{\sigma_m}^{\text{old}}(\sigma_{m'}) = \sum_{n=2}^{N} \eta_n^{\text{old}}(\sigma_m, \sigma_{m'}), \qquad m' = 1, \ldots, M$$

Maximization for emission parameters Unlike the first two optimizations, the third one generally cannot be solved analytically. Instead, depending on the functional form of the density $G(w; \phi)$, numerical technics are needed for the computation of the improved values $\phi_{\sigma_m}^{\text{new}}$ of the optimizers $\phi_{\sigma_m}^{*}$ we are after. Nevertheless, as we show on example 8.3, emission distributions $\mathbb{G}_\phi$ in the exponential family lead to tractable maximizations without numerical optimization.

Example 8.3: Estimation in a HMM with Normal observations

Consider a HMM with state-space $\sigma_{1:M}$ and normal emissions

$$G\left(w; \mu_{\sigma_m}, v_{\sigma_m}\right) = \text{Normal}\left(w; \mu_{\sigma_m}, v_{\sigma_m}\right)$$

where the state parameters are $\phi_{\sigma_m} = (\mu_{\sigma_m}, v_{\sigma_m})$. Further, suppose that an approximation $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ of the maximizer $\boldsymbol{\rho}^{*}, \boldsymbol{\Pi}^{*}, \boldsymbol{\phi}^{*}$ has been already computed and we seek an improved one $\boldsymbol{\rho}^{\text{new}}, \boldsymbol{\Pi}^{\text{new}}, \boldsymbol{\phi}^{\text{new}}$ through the Baum-Welch method.

Due to the special form of $G(w; \mu_{\sigma_m}, v_{\sigma_m})$ in this example, we can work the maximization of the emission parameters analytically. In detail, for each σ_m, the improved emission parameters $\mu_{\sigma_m}^{\text{new}}, v_{\sigma_m}^{\text{new}}$ maximize

$$\sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \log G\left(w_n; \mu_{\sigma_m}, v_{\sigma_m}\right) \propto \sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \left(-\log v_{\sigma_m} - \frac{\left(w_n - \mu_{\sigma_m}\right)^2}{v_{\sigma_m}} \right).$$

Since the maximizers are critical points of this objective, they are found by solving

$$\frac{\partial}{\partial \mu_{\sigma_m}} \sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \left(-\log v_{\sigma_m} - \frac{\left(w_n - \mu_{\sigma_m}\right)^2}{v_{\sigma_m}} \right) = 0$$
$$\frac{\partial}{\partial v_{\sigma_m}} \sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \left(-\log v_{\sigma_m} - \frac{\left(w_n - \mu_{\sigma_m}\right)^2}{v_{\sigma_m}} \right) = 0$$

The solution is

$$\mu_{\sigma_m}^{\text{new}} = \frac{\sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) w_n}{\sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m)}, \qquad v_{\sigma_m}^{\text{new}} = \frac{\sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \left(w_n - \mu_{\sigma_m}^{\text{new}}\right)^2}{\sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m)}.$$

8.3.4 Some computational considerations*

As we have seen already, the forward $\mathcal{A}_n(s_n)$ and backward $\mathcal{B}_n(s_n)$ terms are central to nearly every algorithm we have encountered so far and their accurate evaluation is essential. Unfortunately, the computations in algorithms 8.1 and 8.2, which rely on the recursions of eqs. (*8.10*) and (*8.15*), involve a large number of multiplications between small numbers. Consequently, these algorithms are of limited practical value as most often they lead to rapid *underflow* and erroneous results.

In practice, underflow is prevented if we consider *normalized* forward and backward terms

$$\hat{\mathcal{A}}_n(s_n) = p(s_n|w_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}), \qquad \hat{\mathcal{B}}_n(s_n) = \frac{p\left(w_{n+1:N}|s_n, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)}{p\left(w_{n+1:N}|w_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)}$$

and perform the recursions for $\hat{\mathcal{A}}_n(s_n)$ and $\hat{\mathcal{B}}_n(s_n)$ instead of $\mathcal{A}_n(s_n)$ and $\mathcal{B}_n(s_n)$. In these cases, the recursions needed rely on

$$\hat{\mathcal{A}}_1(s_1) = \frac{1}{\hat{\mathcal{C}}_1} G\left(w_1; \phi_{s_1}\right) \rho_{s_1} \tag{8.28}$$

$$\hat{\mathcal{A}}_n(s_n) = \frac{1}{\hat{\mathcal{C}}_n} G\left(w_n; \phi_{s_n}\right) \sum_{s_{n-1}} \pi_{s_{n-1}\to s_n} \hat{\mathcal{A}}_{n-1}(s_{n-1}), \qquad n = 2, \ldots, N \tag{*8.29*}$$

$$\hat{\mathcal{B}}_n(s_n) = \frac{1}{\hat{\mathcal{C}}_{n+1}} \sum_{s_{n+1}} \hat{\mathcal{B}}_{n+1}(s_{n+1}) G\left(w_{n+1}; \phi_{s_{n+1}}\right) \pi_{s_n\to s_{n+1}}, \qquad n = 1, \ldots, N-1 \tag{*8.30*}$$

$$\hat{\mathcal{B}}_N(s_N) = 1 \tag{8.31}$$

with the constants $\hat{\mathcal{C}}_n$ given by

$$\begin{aligned} \hat{\mathcal{C}}_1 &= p(w_1|\boldsymbol{\rho}, \boldsymbol{\phi}) \\ \hat{\mathcal{C}}_n &= p(w_n|w_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}), \qquad n = 2, \ldots, N. \end{aligned}$$

Because, for each time level, the terms $\hat{\mathcal{A}}_n(\sigma_1), \ldots, \hat{\mathcal{A}}_n(\sigma_M)$ are valid probabilities themselves, they are already scaled self-consistently and underflow is avoided. Further, because $\sum_{s_n} \hat{\mathcal{A}}_n(s_n) = 1$, the constants $\hat{\mathcal{C}}_n$, can be easily computed during the forward recursion. The steps involved in both recursions are summarized in algorithms 8.5 and 8.6.

*This is an advanced topic and could be skipped on a first reading.

Algorithm 8.5: Forward recursion for HMM (stable version)

Given observations $w_{1:N}$ and parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ the forward terms $\hat{\mathcal{A}}_n(\sigma_m)$, for each time level n and each state σ_m, are computed as follows:

- At $n = 1$ initialize by

$$\hat{\mathcal{A}}'_1(\sigma_m) = G(w_1; \phi_{\sigma_m})\rho_{\sigma_m}, \qquad m = 1, \ldots, M$$
$$\hat{\mathcal{C}}_1 = \sum_{\sigma_m} \hat{\mathcal{A}}'_n(\sigma_m)$$
$$\hat{\mathcal{A}}_n(\sigma_m) = \frac{1}{\hat{\mathcal{C}}_1}\hat{\mathcal{A}}'_n(\sigma_m), \qquad m = 1, \ldots, M$$

- For $n = 2, \ldots, N$ compute recursively

$$\mathcal{A}'_n(\sigma_m) = G(w_n; \phi_{\sigma_m}) \sum_{\sigma_{m'}} \pi_{\sigma_{m'} \to \sigma_m} \hat{\mathcal{A}}_{n-1}(\sigma_{m'}), \qquad m = 1, \ldots, M$$
$$\hat{\mathcal{C}}_n = \sum_{\sigma_m} \hat{\mathcal{A}}'_n(\sigma_m)$$
$$\hat{\mathcal{A}}_n(\sigma_m) = \frac{1}{\hat{\mathcal{C}}_n}\hat{\mathcal{A}}'_n(\sigma_m), \qquad m = 1, \ldots, M$$

Upon completion, the algorithm provides every $\hat{\mathcal{A}}_n(\sigma_m)$ and $\hat{\mathcal{C}}_n$ which may be tabulated as

$$\begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_n \\ \vdots \\ t_N \end{array} \begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \hat{\mathcal{A}}_1(\sigma_1) & \hat{\mathcal{A}}_1(\sigma_2) & \cdots & \hat{\mathcal{A}}_1(\sigma_M) \\ \hat{\mathcal{A}}_2(\sigma_1) & \hat{\mathcal{A}}_2(\sigma_2) & \cdots & \hat{\mathcal{A}}_2(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathcal{A}}_n(\sigma_1) & \hat{\mathcal{A}}_n(\sigma_2) & \cdots & \hat{\mathcal{A}}_n(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathcal{A}}_N(\sigma_1) & \hat{\mathcal{A}}_N(\sigma_2) & \cdots & \hat{\mathcal{A}}_N(\sigma_M) \end{bmatrix} \end{array} \begin{bmatrix} \hat{\mathcal{A}}_1(s_1) \\ \hat{\mathcal{A}}_2(s_2) \\ \vdots \\ \hat{\mathcal{A}}_n(s_n) \\ \vdots \\ \hat{\mathcal{A}}_N(s_N) \end{bmatrix} \begin{bmatrix} \hat{\mathcal{C}}_1 \\ \hat{\mathcal{C}}_2 \\ \vdots \\ \hat{\mathcal{C}}_n \\ \vdots \\ \hat{\mathcal{C}}_N \end{bmatrix}$$

Algorithm 8.6: Backward recursion for HMM (stable version)

Given observations $w_{1:N}$, parameters $\boldsymbol{\Pi}, \boldsymbol{\phi}$ and $\hat{\mathcal{C}}_{2:N}$, the backward terms $\mathcal{B}_n(\sigma_m)$, for each time level n and each state σ_m, are computed as follows:

- At $n = N$ initialize by

$$\hat{\mathcal{B}}_N(\sigma_m) = 1, \qquad m = 1, \ldots, M$$

- For $n = N-1, \ldots, 1$ compute recursively

$$\hat{\mathcal{B}}_n(\sigma_m) = \frac{1}{\hat{\mathcal{C}}_{n+1}} \sum_{\sigma_{m'}} \hat{\mathcal{B}}_{n+1}(\sigma_{m'}) G\left(w_{n+1}; \phi_{\sigma_{m'}}\right) \pi_{\sigma_m \to \sigma_{m'}}, \qquad m = 1, \ldots, M$$

Upon completion, the algorithm provides every $\hat{\mathcal{B}}_n(\sigma_m)$ which may be tabulated as

$$\begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_n \\ \vdots \\ t_N \end{array} \begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \hat{\mathcal{B}}_1(\sigma_1) & \hat{\mathcal{B}}_1(\sigma_2) & \cdots & \hat{\mathcal{B}}_1(\sigma_M) \\ \hat{\mathcal{B}}_2(\sigma_1) & \hat{\mathcal{B}}_2(\sigma_2) & \cdots & \hat{\mathcal{B}}_2(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathcal{B}}_n(\sigma_1) & \hat{\mathcal{B}}_n(\sigma_2) & \cdots & \hat{\mathcal{B}}_n(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathcal{B}}_N(\sigma_1) & \hat{\mathcal{B}}_N(\sigma_2) & \cdots & \hat{\mathcal{B}}_N(\sigma_M) \end{bmatrix} \end{array} \begin{array}{c} \\ \hat{\mathcal{B}}_1(s_1) \\ \hat{\mathcal{B}}_2(s_2) \\ \vdots \\ \hat{\mathcal{B}}_n(s_n) \\ \vdots \\ \hat{\mathcal{B}}_N(s_N) \end{array}$$

As we can see, algorithms 8.1 and 8.2 involve more computations than algorithms 8.5 and 8.6; nevertheless, this difference is almost negligible since in both versions the most expensive operation is a matrix-vector multiplication which scales with $M^2 N$. In any case, although somewhat less efficient than $\mathcal{A}_n(s_n), \mathcal{B}_n(s_n)$, computing $\hat{\mathcal{A}}_n(s_n), \hat{\mathcal{B}}_n(s_n)$ avoids underflow which is a serious advantage. Therefore, computing the normalized terms is preferable, especially for large problems. Furthermore, because

$$\hat{\mathcal{A}}_n(s_n) = \mathcal{A}_n(s_n) \frac{1}{p(w_{1:n}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} \tag{*8.32*}$$

$$\hat{\mathcal{B}}_n(s_n) = \mathcal{B}_n(s_n) \frac{1}{p\left(w_{n+1:N}|w_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)} \tag{*8.33*}$$

the normalized terms $\hat{\mathcal{A}}_n(s_n), \hat{\mathcal{B}}_n(s_n)$ can be practically used anywhere $\mathcal{A}_n(s_n), \mathcal{B}_n(s_n)$ are required. For example, both ways of *decoding* a HMM, *e.g.* eqs. (*8.12*) and (*8.13*) or eqs. (*8.17*) and (*8.18*), are unaffected by the normalization. Similarly, the maximization of eq. (8.23) for *estimating* a HMM, *e.g.* eqs. (8.24) and (8.25), remains unaffected too.

However, an important exception occurs when *evaluating* a HMM. In particular, because eq. (8.8) depends explicitly upon $\mathcal{A}_N(s_N)$, the normalization *does* have an effect and the marginal likelihood $p(w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$ needs to be evaluated differently. The most convenient way is through

$$p(w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) = \prod_{n=1}^{N} \hat{\mathcal{C}}_n \tag{*8.34*}$$

with the constants $\hat{\mathcal{C}}_n$ computed, most efficiently, with the forward recursion of algorithm 8.5.

Note 8.11: Terminology

Due to their importance, the terms $\mathcal{A}_n(s_n)$ and $\mathcal{B}_n(s_n)$ or their normalized counterparts $\hat{\mathcal{A}}_n(s_n)$ and $\hat{\mathcal{B}}_n(s_n)$ appear often across multiple disciplines including Dynamical Systems, Signal Processing, Machine Learning, and Statistics. Generally, the forward terms $\mathcal{A}_n(s_n), \hat{\mathcal{A}}_n(s_n)$ are occasionally termed *filters* or *forward variables* or *forward messages*; while, the backward terms $\mathcal{B}_n(s_n), \hat{\mathcal{B}}_n(s_n)$ are occasionally termed *smoothers* or *backward variables* or *backward messages*. Unfortunately, the terminology is not standardized and remains quite inconsistent. This inconsistency often leads to considerable confusion whenever the full probabilistic descriptions are not provided.

8.3.5 State-space labeling and likelihood invariance*

The algorithms we presented so far are routinely used to answer questions pertaining to a HMM. These algorithms exhibit maximum efficiency for their tasks; however, they are limited to yielding point estimates *only*. That is, at best these algorithms can provide a single choice for the values of the variables of interest, for example $s_n^*, s_{1:N}^\sharp$ or $\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*$, that is optimal under certain criteria. Unfortunately, they fail to quantify the uncertainty associated with each estimator, which is a serious limitation by itself.

Error bars around the estimators may be obtained with generic likelihood-based strategies, for example through Fisher information or bootstrapping. Indeed, such approaches are possible, at least in theory, under Monte Carlo sampling or greedy computations where each one, or at least a good portion, of in total N^M possible sequences $s_{1:N}$ is exhaustively computed. However, even with greedy computations, there is a fundamental degeneracy in eqs. (8.5) to (8.7) that prohibits the uniqueness of any computed estimator.

Namely, a HMM's likelihoods, $p(w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$ or $p(w_{1:N}, s_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$, are *invariant to permutations* of the constituting state labels. That is, relabeling of the constituting states results in the same value of the likelihood. Consequently, irrespective of how an estimator is obtained, there are always additional $M! - 1$ equally optimal ones.

Example 8.4: State relabeling

To illustrate the degeneracy of the likelihood, we consider a simplified HMM containing $N = 3$ time levels and $M = 2$ constituting states. Further, for clarity we adopt pedantic notation and let $s_{1:3}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ stand for the corresponding random variables. Considering realized values for these random variables, invariance of the (marginal) likelihood reads

$$\begin{aligned} &p\left(w_{1:3} \,\middle|\, \boldsymbol{\rho} = (\rho_\alpha, \rho_\beta), \boldsymbol{\Pi} = \begin{pmatrix} \pi_{\alpha\to\alpha} & \pi_{\alpha\to\beta} \\ \pi_{\beta\to\alpha} & \pi_{\beta\to\beta} \end{pmatrix}, \boldsymbol{\phi} = (\phi_\alpha, \phi_\beta)\right) \\ &= p\left(w_{1:3} \,\middle|\, \boldsymbol{\rho} = (\rho_\beta, \rho_\alpha), \boldsymbol{\Pi} = \begin{pmatrix} \pi_{\beta\to\beta} & \pi_{\beta\to\alpha} \\ \pi_{\alpha\to\beta} & \pi_{\alpha\to\alpha} \end{pmatrix}, \boldsymbol{\phi} = (\phi_\beta, \phi_\alpha)\right). \end{aligned}$$

Similarly, invariance of the (joint) likelihood reads

$$\begin{aligned} &p\left(w_{1:3}, s_{1:3} = (\alpha,\beta,\beta) \,\middle|\, \boldsymbol{\rho} = (\rho_\alpha, \rho_\beta), \boldsymbol{\Pi} = \begin{pmatrix} \pi_{\alpha\to\alpha} & \pi_{\alpha\to\beta} \\ \pi_{\beta\to\alpha} & \pi_{\beta\to\beta} \end{pmatrix}, \boldsymbol{\phi} = (\phi_\alpha, \phi_\beta)\right) \\ &= p\left(w_{1:3}, s_{1:3} = (\beta,\alpha,\alpha) \,\middle|\, \boldsymbol{\rho} = (\rho_\beta, \rho_\alpha), \boldsymbol{\Pi} = \begin{pmatrix} \pi_{\beta\to\beta} & \pi_{\beta\to\alpha} \\ \pi_{\alpha\to\beta} & \pi_{\alpha\to\alpha} \end{pmatrix}, \boldsymbol{\phi} = (\phi_\beta, \phi_\alpha)\right). \end{aligned}$$

Both cases are produced by considering every possible permutation of the constituting states, which for this simple example are

$$\begin{matrix} \sigma_1 & \sigma_2 \\ \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix}. \end{matrix}$$

*This is an advanced topic and could be skipped on a first reading.

Had our HMM a larger state-space, the total number of label permutations possible would be larger. For instance, with $M = 3$, the likelihoods have a 6-fold degeneracy produced by

$$\begin{matrix} \sigma_1 & \sigma_2 & \sigma_2 \end{matrix} \\ \begin{pmatrix} \alpha & \beta & \gamma \\ \alpha & \gamma & \beta \\ \beta & \alpha & \gamma \\ \beta & \gamma & \alpha \\ \gamma & \alpha & \beta \\ \gamma & \beta & \alpha \end{pmatrix}$$

In general, a state space of size M, entails an $M!$-fold degeneracy.

In other words, the HMM of eqs. (8.5) to (8.7) is not identifiable in the *strict* sense where unique estimators are associated with each label m. Instead it is identifiable only up to permutations of the state labels. As we are mostly interested in the association between constitutive or occupying states and parameters, in practice, such non-identifiability does not pose a problem. For example, the sequence of emission parameters $\phi_{s_1} \to \cdots \to \phi_{s_N}$ successively attained by the system at hand does not depend on a particular labeling and so it is uniquely identifiable.

Note 8.12

The degeneracy of the likelihoods does not impact the execution of the algorithms developed so far. For instance, algorithms 8.1 to 8.3 or algorithms 8.5 and 8.6 assume already known parameters which presume a pre-existent labeling of the state-space $\sigma_{1:M}$. Similarly, a pre-existing labeling of the state-space $\sigma_{1:M}$ is also assumed in algorithm 8.4.

In essence, the likelihood's invariance under relabeling originates in $p(w_{1:N}|s_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$ and occurs because dynamic and emission parameters are defined *only* with regards to the constituting states σ_m and *not* their numerical labels m. To eliminate the HMM's invariance, assumptions *additional* to those listed on section 8.2 are required in order to establish an association between constitutive states σ_m and their labels m.

In practice, we can resolve a HMM non-identifiability by imposing *post hoc* identifiably constrains in terms of the state labels. For example, once an estimator is chosen, error bars can be obtained by restricting the entire parameter space only to the semi-orthant that this estimator lies on. In this case, heuristic rules, that are problem specific, are needed to impose a particular relation of the constitutive states with their assigned numerical labels. For instance, restricting to a particular semi-orthant of the parameter space entails that from all $M!$ equally likely relabelings we select this one that lies closest to the *ad hoc* labeling of the estimator. Because this scenario is most prominent in the Bayesian context, we discuss it in more detail in the next section.

8.4 The Hidden Markov Model in the Bayesian paradigm

As we have seen in previous chapters, a Bayesian formulation provides more modeling flexibility than its frequentist counterpart. In practice, such flexibility is quite useful when modeling dynamical systems where important information, that *cannot* be provided by the measurements alone, needs to be incorporated to aid inference. For instance, we see shortly, with a Bayesian HMM we can influence time scales for the kinetics or prescribe kinetic schemes. As we also see in later sections, among other unique features, we can generalize a Bayesian HMM to account for uncharacterized state-spaces too.

In the Bayesian setting we rely on posterior estimates. Therefore, every question about a system formulated with a Bayesian HMM is answered through the corresponding posterior $p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$ or the completed posterior $p(s_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$. The HMM of eqs. (8.5) to (8.7) provides probability distributions only for the occupying states $p(s_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi})$ and the measurements $p(w_{1:N}|s_{1:N}, \boldsymbol{\phi})$, which do not suffice to fully specify the posteriors we are after. For this reason, a Bayesian HMM requires the specification of additional distributions that supply

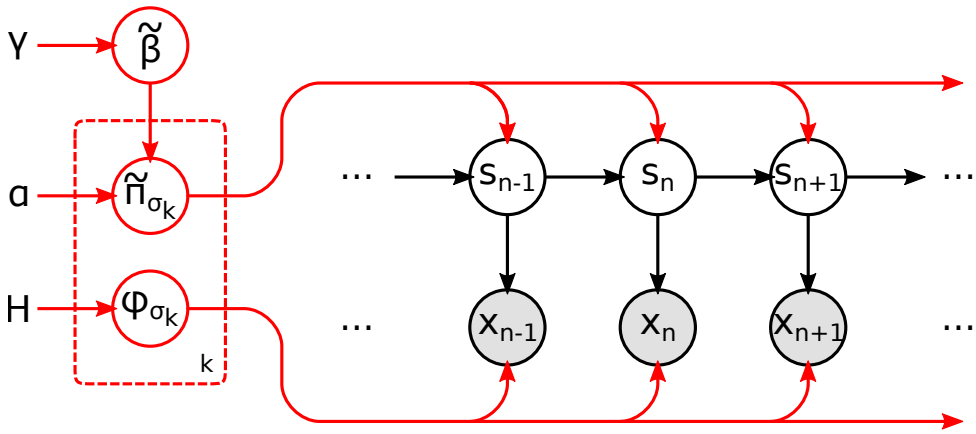


Figure 8.3: Graphical representation of a Bayesian hidden Markov model. Here, observations x_n are dark shaded and parameters $\tilde{\pi}_{\sigma_m}$ and ϕ_{σ_m} are modeled as random variables.

statistics to the parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$. These are our priors and, as anticipated, several reasonable choices can be devised to accommodate a system at hand.

Below, we describe suitable prior choices and subsequently appropriate sampling techniques for a plain HMM. That is, a generic HMM formulated in Bayesian terms. We present specialized versions, tailored to specific cases, in the next sections. As in section 8.3, to increase clarity, we omit lengthy derivations of key equations in this section. We mark these equations with ** and list their derivation in appendix E.

8.4.1 Priors for the HMM

The simplest choice for the initial ρ_{σ_m} and transition probabilities $\pi_{\sigma_m \to \sigma_{m'}}$ are offered by placing independent priors on $\boldsymbol{\rho}$ and each $\boldsymbol{\pi}_{\sigma_m}$. For instance, Dirichlet distributions

$$\begin{aligned} \boldsymbol{\rho} &\sim \text{Dirichlet}_M\left(\eta\boldsymbol{\zeta}\right) \\ \boldsymbol{\pi}_{\sigma_m} &\sim \text{Dirichlet}_M\left(\alpha_{\sigma_m}\boldsymbol{\beta}_{\sigma_m}\right), \qquad\qquad m = 1, \ldots, M \end{aligned}$$

ensure our parameters are valid probably arrays. In these priors, η and α_{σ_m} are positive scalar constants; while, $\boldsymbol{\zeta} = [\zeta_{\sigma_1}, \ldots, \zeta_{\sigma_M}]$ and $\boldsymbol{\beta}_{\sigma_m} = [\beta_{\sigma_m \to \sigma_1}, \ldots, \beta_{\sigma_m \to \sigma_1}]$ are probability arrays. Due to the conjugacy between the Categorical and Dirichlet distributions, as we will see shortly, such choices facilitate computational tractability.

Despite the generality of the dynamical parameters, a choice for the emission parameters ϕ_{σ_m} depends heavily on the emission distributions $\mathbb{G}_{\phi_{\sigma_m}}$ which, in turn, vary widely between systems. Generally, computational tractability is facilitated if we consider iid priors

$$\phi_{\sigma_m} \sim \mathbb{H}, \qquad\qquad m = 1, \ldots, M \tag{8.35}$$

under a common probably distribution $\mathbb{H}$. Additionally, we see below that the computations involved are greatly simplified, if the probability density $H(\phi)$ of $\mathbb{H}$ is conjugate to the probability density $G(w; \phi)$ of the mother distribution $\mathbb{G}_\phi$.

8.4.2 MCMC inference in the Bayesian HMM

With the choices described above, the entire Bayesian HMM is summarized in

$$\boldsymbol{\rho} \sim \text{Dirichlet}_M\left(\eta\boldsymbol{\zeta}\right), \tag{8.36}$$

$$\boldsymbol{\pi}_{\sigma_m} \sim \text{Dirichlet}_M\left(\alpha_{\sigma_m}\boldsymbol{\beta}_{\sigma_m}\right), \qquad m = 1, \ldots, M \tag{8.37}$$

$$\phi_{\sigma_m} \sim \mathbb{H}, \qquad m = 1, \ldots, M \tag{8.38}$$

$$s_1|\boldsymbol{\rho} \sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\rho}), \tag{8.39}$$

$$s_n|s_{n-1}, \boldsymbol{\Pi} \sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\pi}_{\sigma_m}), \qquad n = 2, \ldots, N \tag{8.40}$$

$$w_n|s_n, \boldsymbol{\phi} \sim \mathbb{G}_{\phi_{s_n}}, \qquad n = 1, \ldots, N \tag{8.41}$$

and illustrated in fig. 8.3. Generally, inference in this HMM is more complicated than in its non-Bayesian counterpart. Below, we describe two complementing MCMC sampling approaches. One is based on the Gibbs sampler, which is sufficient for most applications. The other is based on the Metropolis-Hastings sampler and is deemed for demanding applications where mixing of the Gibbs sampler becomes inefficient.

Gibbs sampling

A Gibbs sampling scheme for generating MCMC samples from the HMM's posterior may be achieved based on the completion

$$p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N}) = \sum_{s_{1:N}} p(s_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N}).$$

In a basic implementation, we need to iterate between successive updates of $s_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, w_{1:N}$ and $(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})|s_{1:N}, w_{1:N}$. Due to the formulation of HMM, the later reduces to independent updates for each parameter. Specifically, once $s_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, w_{1:N}$ is sampled, parameters are updated by sampling $\boldsymbol{\rho}|s_1$ and $\boldsymbol{\pi}_{\sigma_m}|s_{1:N}$ and $\phi_{\sigma_m}|s_{1:N}, w_{1:N}$ for each σ_m. The entire scheme is summarized in algorithm 8.7.

Algorithm 8.7: Gibbs sampling for Bayesian HMM

Given an initial sample $\boldsymbol{\rho}^{(0)}, \boldsymbol{\Pi}^{(0)}, \boldsymbol{\phi}^{(0)}$, which may be sampled from the corresponding priors, MCMC updates for the Bayesian HMM are carried out by iterating the steps:

- For $j = 1, 2, \ldots$
  - update the occupying state sequence by sampling $s_{1:N}^{(j)}$ with forward filtering backward sampling based on $\boldsymbol{\rho}^{(j-1)}, \boldsymbol{\Pi}^{(j-1)}, \boldsymbol{\phi}^{(j-1)}$.
  - compute counts $\boldsymbol{d}^{(j)}$ and $\boldsymbol{C}^{(j)}$ based on $s_{1:N}^{(j)}$.
  - update initial and transitions probability vectors by sampling from

$$\boldsymbol{\rho}^{(j)} \sim \text{Dirichlet}_M\left(\eta\boldsymbol{\zeta} + \boldsymbol{d}^{(j)}\right),$$
$$\boldsymbol{\pi}_{\sigma_m}^{(j)} \sim \text{Dirichlet}_M\left(\alpha_{\sigma_m}\boldsymbol{\beta}_{\sigma_m} + \boldsymbol{c}_{\sigma_m}^{(j)}\right), \qquad m = 1, \ldots, M.$$

  - compute state indexes $\mathcal{N}_{\sigma_m}^{(j)}$ based on $s_{1:N}^{(j)}$.
  - update the emission parameters by sampling $\phi_{\sigma_m}^{(j)}$ for each σ_m based on $\mathcal{N}_{\sigma_m}^{(j)}$ and $w_{1:N}$.

Upon completion, the sampler produces MCMC samples $s_{1:N}^{(j)}, \boldsymbol{\rho}^{(j)}, \boldsymbol{\Pi}^{(j)}, \boldsymbol{\phi}^{(j)}$ according to $p(s_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$.

Below, we examine the steps involved in this Gibbs scheme in more detail. For clarity, we designate with $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ a sample in the MCMC chain and with $s_{1:N}^{\text{new}}, \boldsymbol{\rho}^{\text{new}}, \boldsymbol{\Pi}^{\text{new}}, \boldsymbol{\phi}^{\text{new}}$ its very next one.

Updates of the occupying state sequence In the Gibbs sampler, the occupying state sequence is updated by sampling from $p(s_{1:N}|\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}, w_{1:N})$. This distribution may be factorized as

$$p(s_{1:N}|\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}, w_{1:N}) = p(s_N|\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}, w_{1:N}) \prod_{n=1}^{N-1} p(s_n|s_{n+1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}, w_{1:N})$$

which allows s_N^{new} to be sampled fist and subsequently each s_n^{new} to be sampled recursively backwards. We can perform such sampling using the forward terms $\hat{\mathcal{A}}_n(\sigma_m)$ which need to be precomputed through filtering, for example, via algorithm 8.1. Because these terms need to be computed under $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ we designate them with $\hat{\mathcal{A}}_n^{\text{old}}(\sigma_m)$.

Precisely, once every $\hat{\mathcal{A}}_n^{\text{old}}(\sigma_m)$ is computed with a forward recursion, sampling begins with

$$s_N^{\text{new}} \sim \text{Categorical}_{\sigma_{1:M}}\left(\hat{\mathcal{A}}_N^{\text{old}}(\sigma_1), \cdots, \hat{\mathcal{A}}_N^{\text{old}}(\sigma_M)\right) \tag{*8.42*}$$

and recurses backward based on

$$s_n^{\text{new}} \sim \text{Categorical}_{\sigma_{1:M}}\left(\frac{\hat{\mathcal{A}}_n^{\text{old}}(\sigma_1)\pi^{\text{old}}_{\sigma_1 \to s_{n+1}^{\text{new}}}}{\sum_{\sigma_m}\hat{\mathcal{A}}_n^{\text{old}}(\sigma_m)\pi^{\text{old}}_{\sigma_m \to s_{n+1}^{\text{new}}}}, \cdots, \frac{\hat{\mathcal{A}}_n^{\text{old}}(\sigma_M)\pi^{\text{old}}_{\sigma_M \to s_{n+1}^{\text{new}}}}{\sum_{\sigma_m}\hat{\mathcal{A}}_n^{\text{old}}(\sigma_m)\pi^{\text{old}}_{\sigma_m \to s_{n+1}^{\text{new}}}}\right). \tag{*8.43*}$$

The entire processes is termed *forward filtering backward sampling* and the steps involved are summarized in algorithm 8.8.

Algorithm 8.8: Forward filtering backward sampling

Given observations $w_{1:N}$ and parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$, an occupying state sequence $s_{1:N}$ is sampled as follows:

- Use algorithm 8.5 and $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ to compute $\hat{\mathcal{A}}_n(\sigma_m)$
- At $n = N$ generate

$$s_N \sim \text{Categorical}_{\sigma_{1:M}}\left(\hat{\mathcal{A}}_N(\sigma_1), \cdots, \hat{\mathcal{A}}_N(\sigma_M)\right)$$

- For $n = N-1, \ldots, 1$ generate recursively

$$s_n \sim \text{Categorical}_{\sigma_{1:M}}\left(\frac{\hat{\mathcal{A}}_n(\sigma_1)\pi_{\sigma_1 \to s_{n+1}}}{\sum_{\sigma_m}\hat{\mathcal{A}}_n(\sigma_m)\pi_{\sigma_m \to s_{n+1}}}, \cdots, \frac{\hat{\mathcal{A}}_n(\sigma_M)\pi_{\sigma_M \to s_{n+1}}}{\sum_{\sigma_m}\hat{\mathcal{A}}_n(\sigma_m)\pi_{\sigma_m \to s_{n+1}}}\right)$$

Upon completion, the algorithm provides $s_{1:N}$ distributed according to $p(s_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, w_{1:N})$.

Updates of the dynamic parameters In the Gibbs sampler, the initial probabilities are updated by sampling from $p(\boldsymbol{\rho}|s_1^{\text{new}})$. In turn, due to conjugacy, sampling reduces to

$$\boldsymbol{\rho}^{\text{new}} \sim \text{Dirichlet}_M\left(\eta\boldsymbol{\zeta} + \boldsymbol{d}^{\text{new}}\right) \tag{*8.44*}$$

where $\boldsymbol{d}^{\text{new}} = [d_{\sigma_1}^{\text{new}}, \cdots, d_{\sigma_M}^{\text{new}}]$ is an array of 0 and 1 whose σ_m entry indicates whether $s_1^{\text{new}} = \sigma_m$ or not.

Similarly, the transition probabilities out of each constitutive state σ_m are updated by sampling from $p(\boldsymbol{\pi}_{\sigma_m}|s_{1:N}^{\text{new}})$. Again, due to conjugacy, sampling reduces to

$$\boldsymbol{\pi}_{\sigma_m}^{\text{new}} \sim \text{Dirichlet}_M\left(\alpha_{\sigma_m}\boldsymbol{\beta}_{\sigma_m} + \boldsymbol{c}_{\sigma_m}^{\text{new}}\right) \tag{*8.45*}$$

where $\boldsymbol{c}_{\sigma_m}^{\text{new}} = [c_{\sigma_m \to \sigma_1}^{\text{new}}, \cdots, c_{\sigma_m \to \sigma_M}^{\text{new}}]$ is a vector whose $\sigma_m \to \sigma_{m'}$ entry counts how many times the transition $\sigma_m \to \sigma_{m'}$ occurs in $s_{1:N}^{\text{new}}$.

Note 8.13: Transition count matrix

Bookkeeping is simpler, if we tabulate the count arrays $\boldsymbol{c}_{\sigma_m}$ into

$$\begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_M \end{array} \begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} c_{\sigma_1\to\sigma_1} & c_{\sigma_1\to\sigma_2} & \cdots & c_{\sigma_1\to\sigma_M} \\ c_{\sigma_2\to\sigma_1} & c_{\sigma_2\to\sigma_2} & \cdots & c_{\sigma_2\to\sigma_M} \\ \vdots & \vdots & \ddots & \vdots \\ c_{\sigma_M\to\sigma_1} & c_{\sigma_M\to\sigma_2} & \cdots & c_{\sigma_M\to\sigma_M} \end{bmatrix} \end{array} = \begin{bmatrix} \boldsymbol{c}_{\sigma_1} \\ \boldsymbol{c}_{\sigma_2} \\ \vdots \\ \boldsymbol{c}_{\sigma_M} \end{bmatrix} = \boldsymbol{C}$$

which is similar to the tabulation of $\boldsymbol{\Pi}$.

Updates of the observation parameters In the Gibbs sampler, the emission parameters of each constituting state σ_m are updated by sampling from $p(\phi_{\sigma_m}|s^{\text{new}}_{1:N}, w_{1:N})$. Using Bayes's rule, this distribution factorizes as

$$p(\phi_{\sigma_m}|s^{\text{new}}_{1:N}, w_{1:N}) \propto H(\phi_{\sigma_m}) \prod_{n\in\mathcal{N}^{\text{new}}_{\sigma_m}} G(w_n;\phi_{\sigma_m}) \tag{*8.46*}$$

where $\mathcal{N}^{\text{new}}_{\sigma_m}$ gathers the indexes n of the time levels such that $s^{\text{new}}_n = \sigma_m$.

For arbitrary $\mathbb{H}$ and $\mathbb{G}_\phi$, this sampling cannot be performed directly and a Metropolis-within-Gibbs scheme is required. Nevertheless, as we show on **??**, distributions $\mathbb{G}_\phi$ in the exponential family, with conjugate priors $\mathbb{H}$, can be sampled directly.

Metropolis-Hastings sampling*

The Gibbs sampler described so far is most often sufficient for applications of the HMM, especially whenever the total number of time levels N is low or the emission distributions $\mathbb{F}_{\sigma_m}$ overlap appreciably. However, for long sequences and/or well separated emission distributions mixing of the Gibbs sampler may become particularly poor. For such cases, an alternative sampler that updates $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ while keeping the occupying state sequence $s_{1:N}$ marginalized is preferable.

Such a sampler may be developed on the same principles as the generic Metropolis-Hastings sampler. In particular, to sample from a HMM's posterior $p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$ a Metropolis-Hastings sampler requires selecting a suitable proposal $Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}})$. Although such proposal may attempt to update all parameters at once, in general, it is more practical to update one or at most few parameters at a time. This may be achieved by a mixture proposal, for example, of the form

$$\begin{aligned} Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}) = \omega\, Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}}}(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}})\, \delta_{\boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\phi}^{\text{prop}}) \\ + (1-\omega)\, Q_{\boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\phi}^{\text{prop}})\, \delta_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}}}(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}) \end{aligned}$$

where with probability ω only proposals for the dynamic parameters are made; while, with probability $1-\omega$ only proposals for the emission parameters are made. In turn each of the partial proposals $Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}}}(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}})$ and $Q_{\boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\phi}^{\text{prop}})$ may consist of further mixtures themselves that propose $\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\pi}^{\text{prop}}_{\sigma_m}, \phi^{\text{prop}}_{\sigma_m}$ separately.

Note 8.14: Choice of proposals

Generally, $Q_{\boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\phi}^{\text{prop}})$ is problem specific; however, for $Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}}}(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}})$ we may construct generic proposals by considering products of Dirichlet distributions. For example as

$$Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}}}(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}) = \text{Dirichlet}_M\left(\boldsymbol{\rho}^{\text{prop}}; \kappa\boldsymbol{\rho}^{\text{old}}\right) \prod_{\sigma_m} \text{Dirichlet}_M\left(\boldsymbol{\pi}^{\text{prop}}_{\sigma_m}; \lambda\boldsymbol{\pi}^{\text{old}}_{\sigma_m}\right)$$

*This is an advanced topic and could be skipped on a first reading.

This choice ensures that the proposed $\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{old}}$ consist of valid probability arrays and also allows for tuning of the resulting acceptance rate through the parameters κ and λ which control how $\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}$ are spread around $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}$.

Finally, once a proposal $\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}$ is made, either through $Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}\right)$ or $Q_{\boldsymbol{\phi}^{\text{old}}}\left(\boldsymbol{\phi}^{\text{prop}}\right)$, an acceptance ratio

$$R_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}\right) = \frac{p\left(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}} | w_{1:N}\right)}{p\left(\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}} | w_{1:N}\right)} \times \frac{Q_{\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}}\left(\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right)}{Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}\right)}$$

must be computed to complete the Metropolis-Hastings acceptance test. The second ratio depends on the specific choices for the proposals made and can be easily computed. The first ratio, however, requires marginalizing over $s_{1:N}$ and this may be achieved through the factorizations

$$\frac{p\left(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}} | w_{1:N}\right)}{p\left(\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}} | w_{1:N}\right)} = \frac{p\left(w_{1:N} | \boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}\right)}{p\left(w_{1:N} | \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right)} \frac{p\left(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}\right)}{p\left(\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right)}.$$

The last ratio depends exclusively on the HMM's priors and can also be readily computed. The other ratio is formed by the HMM's marginal likelihoods which we can evaluate thought eq. (*8.34*). For this case, a filtering algorithm such as algorithm 8.5 needs to be invoked twice: once for $p\left(w_{1:N} | \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right)$ and once for $p\left(w_{1:N} | \boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}\right)$.

Because, filtering make up the most computationally intensive part in a HMM, both likelihoods can be retained and updated upon acceptance. By doing so, at the next iteration we may avoid the recomputation of $p\left(w_{1:N} | \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right)$, this way, reducing the computational load to invoking *one* filtering per iteration. Such cost is competitive with the Gibbs sampler described earlier. The entire Metropolis-Hastings scheme is summarized in algorithm 8.9.

Algorithm 8.9: Metropolis-Hastings sampling for Bayesian HMM

Given an initial sample $\boldsymbol{\rho}^{(0)}, \boldsymbol{\Pi}^{(0)}, \boldsymbol{\phi}^{(0)}$, which may be sampled from the corresponding priors, MCMC updates for the Bayesian HMM are carried out by:

- Compute $\hat{\mathcal{A}}^{(0)}_{1:N}(\sigma_m)$ and $\hat{\mathcal{C}}^{(0)}_{1:N}$ based on $\boldsymbol{\rho}^{(0)}, \boldsymbol{\Pi}^{(0)}, \boldsymbol{\phi}^{(0)}$.
- Compute $\mathcal{L}^{(0)} = \sum_n \log \hat{\mathcal{C}}^{(0)}_n$.
- For $j = 1, 2, \ldots$
  - Generate proposals $\boldsymbol{\rho}', \boldsymbol{\Pi}', \boldsymbol{\phi}'$ based on $\boldsymbol{\rho}^{(j-1)}, \boldsymbol{\Pi}^{(j-1)}, \boldsymbol{\phi}^{(j-1)}$.
    - Compute $\hat{\mathcal{A}}'_{1:N}(\sigma_m)$ and $\hat{\mathcal{C}}'_{1:N}$ based on $\boldsymbol{\rho}', \boldsymbol{\Pi}', \boldsymbol{\phi}'$.
    - Compute $\mathcal{L}' = \sum_n \log \hat{\mathcal{C}}'_n$.
    - Perform the Metropolis-Hastings acceptance test based on $\mathcal{L}^{(j-1)}, \boldsymbol{\rho}^{(j-1)}, \boldsymbol{\Pi}^{(j-1)}, \boldsymbol{\phi}^{(j-1)}$ and $\mathcal{L}', \boldsymbol{\rho}', \boldsymbol{\Pi}', \boldsymbol{\phi}'$.
    - If acceptance test *succeeds*, set $\boldsymbol{\rho}^{(j)} = \boldsymbol{\rho}', \boldsymbol{\Pi}^{(j)} = \boldsymbol{\Pi}', \boldsymbol{\phi}^{(j)} = \boldsymbol{\phi}'$ and $\mathcal{L}^{(j)} = \mathcal{L}', \hat{\mathcal{A}}^{(j)}_n(\sigma_m) = \hat{\mathcal{A}}'_n(\sigma_m)$.
    - If acceptance test *fails*, set $\boldsymbol{\rho}^{(j)} = \boldsymbol{\rho}^{(j-1)}, \boldsymbol{\Pi}^{(j)} = \boldsymbol{\Pi}^{(j-1)}, \boldsymbol{\phi}^{(j)} = \boldsymbol{\phi}^{(j-1)}$ and $\mathcal{L}^{(j)} = \mathcal{L}^{(j-1)}, \hat{\mathcal{A}}^{(j)}_n(\sigma_m) = \hat{\mathcal{A}}^{(j-1)}_n(\sigma_m)$.
  - Update the occupying state sequence by sampling $s^{(j)}_{1:N}$ with backward sampling based on $\hat{\mathcal{A}}^{(j)}_n(\sigma_m)$ and $\boldsymbol{\Pi}^{(j-1)}$.

Computation of $\hat{\mathcal{A}}_n(\sigma_m)$ and $\hat{\mathcal{C}}_n$ is achieved by algorithm 8.5, backward sampling of $s^{(j)}_{1:N}$ is achieved by algorithm 8.8, and the acceptance test is executed by generating $u \sim \text{Uniform}_{[0,1]}$ and testing whether

$$\mathcal{L}' - \mathcal{L}^{(j-1)} + \log \frac{p\left(\boldsymbol{\rho}', \boldsymbol{\Pi}', \boldsymbol{\phi}'\right)}{p\left(\boldsymbol{\rho}^{(j-1)}, \boldsymbol{\Pi}^{(j-1)}, \boldsymbol{\phi}^{(j-1)}\right)} \frac{Q_{\boldsymbol{\rho}', \boldsymbol{\Pi}', \boldsymbol{\phi}'}\left(\boldsymbol{\rho}^{(j-1)}, \boldsymbol{\Pi}^{(j-1)}, \boldsymbol{\phi}^{(j-1)}\right)}{Q_{\boldsymbol{\rho}^{(j-1)}, \boldsymbol{\Pi}^{(j-1)}, \boldsymbol{\phi}^{(j-1)}}\left(\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*\right)} < \log u.$$

Upon completion, the sampler produces MCMC samples $\boldsymbol{\rho}^{(j)}, \boldsymbol{\Pi}^{(j)}, \boldsymbol{\phi}^{(j)}$ according to $p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi} | w_{1:N})$.

Note 8.15: Sampling the occupying state sequence

If needed, algorithm 8.9 can also sample $s_{1:N}$ at little additional cost. For instance, following each filtering, if instead of maintaining only the marginal likelihood we maintain and update also every forward term $\hat{\mathcal{A}}_{1:N}(\sigma_m)$, a new state sequence $s^{\text{new}}_{1:N}$ can be obtained at the end of each Metropolis-Hastings iteration by executing only the backward sampling stage of algorithm 8.8.

Essentially, by maintaining $\boldsymbol{\rho}^{\text{new}}, \boldsymbol{\Pi}^{\text{new}}, \boldsymbol{\phi}^{\text{new}}$ and $\hat{\mathcal{A}}^{\text{new}}_{1:N}(\sigma_m), \hat{\mathcal{C}}^{\text{new}}_{1:N}$ we may efficiently obtain samples from the HMM competed posterior $p(s_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi} | w_{1:N})$.

8.4.3 Interpretation and label switching*

Similar to the HMM's likelihoods we saw in section 8.3.5, the posteriors (marginal or completed) of the Bayesian HMM in eqs. (8.36) to (8.41) are also invariant to label permutations. In this case, the invariance stems from the factorization

$$\begin{aligned} p\left(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi} | w_{1:N}\right) \propto\ & p\left(w_{1:N} | \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) \\ & \times \text{Dirichlet}_M\left(\boldsymbol{\rho}; \eta\boldsymbol{\zeta}\right) \prod_{\sigma_m} \text{Dirichlet}_M\left(\boldsymbol{\pi}_{\sigma_m}; \alpha_{\sigma_m}\boldsymbol{\beta}_{\sigma_m}\right) \prod_{\sigma_m} H(\phi_{\sigma_m}) \end{aligned}$$

and is caused by both the invariance of the likelihood as well as of the priors specified in section 8.4.1.

*This is an advanced topic and could be skipped on a first reading.

Note 8.16: Breaking the posterior's invariance

Unlike the frequentist HMM, in a Bayesian HMM we may avoid the posterior's invariance if we assign priors on the parameters that are *label specific*. For instance, an alternative Bayesian HMM may be constructed with the following choices

$$\begin{aligned} \boldsymbol{\rho} &\sim \text{Dirichlet}_M\left(\eta(\zeta_1,\cdots,\zeta_M)\right), \\ \boldsymbol{\pi}_{\sigma_m} &\sim \text{Dirichlet}_M\left(\alpha_m(\beta_{m\to 1},\cdots,\beta_{m\to M})\right), && m=1,\ldots,M \\ \phi_{\sigma_m} &\sim \mathbb{H}_m, && m=1,\ldots,M \end{aligned}$$

In this version, priors are label specific and as a result relabeling of the state-space leads to different posterior values to each one of the $M!$ samples produced by every possible label permutation. This way, we need not invoke *post hoc* heuristics to resolve identifiability problems.

In practice, it is better if label specific priors are avoided. This is because priors, that are informative on the state labels, may hinder the mixing of the MCMC samplers applied. For best computational efficiency, it is preferable to use priors that are state, and not label, specific.

The posterior's invariance to label permutations leads to multimodal posteriors. For example, for any MAP estimate $\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*$, there are $M!-1$ additional maximizers produced by the label permutations. Each one of these $M!$ maximizers is a local mode of the posterior and is associated with a unique labeling.

As eqs. (8.36) to (8.41) do not show preference for a particular labeling of the state space, in general, MCMC samplers produce samples that use any of them. In fact, a sampler that performs well samples the entire posterior and so, in the long run, switches between them producing samples from all $M!$ posterior modes.

As long as we are interested in deriving estimates that depend only on the constitutive states and not on the particular labeling chosen, the MCMC chains generated are sufficient. For example, if all we care about is quantifying the emission parameters attained at a particular level, we need focus on $p(\phi_{s_n}|w_{1:N})$, which is independent of the state space's labeling.

To derive label specific estimates, and therefore to allow for full interpretation of our estimates, we need to impose *post hoc* identifiability constraints in terms of the state labels similar to the frequentist HMM of section 8.3.5. For instance, because all $M!$ modes are equally probable, for each MCMC sample computed we can consider all other $M!-1$ ones, by forming every possible permutation, and selecting the one that satisfies our constraints. Below we explain in more detail the steps involved.

We suppose that an MCMC sampler has been already employed, and for clarity, we denote with $\boldsymbol{\theta}_k^{(j)} = (\boldsymbol{s}_k^{(j)}, \boldsymbol{\rho}_k^{(j)}, \boldsymbol{\Pi}_k^{(j)}, \boldsymbol{\phi}_k^{(j)})$ and $k=1$ the values sampled at the j^{th} iteration. Further, we use $k>1$ to denote every other sample value that can be formed by $\boldsymbol{\theta}_1^{(j)}$ through permutations of the state labels. Because there are in total $M!$ permutations, we have in total $K=M!$ posterior samples distinguished among each other by $k=1,\cdots,K$. As we mentioned above, due to label invariance, all these samples are equiprobable

$$p\left(\boldsymbol{\theta}_1^{(j)}|w_{1:N}\right) = p\left(\boldsymbol{\theta}_k^{(j)}|w_{1:N}\right), \qquad k=1,\ldots,K.$$

To restore identifiability, it is sufficient if we select a single $\boldsymbol{\theta}_k^{(j)}$ out of $\boldsymbol{\theta}_{1:K}^{(j)}$ that satisfies our constrains. Because the permutation that satisfies the constrains generally may differ from iteration to iteration, we designate it with $k^{(j)}$.

Perhaps the simplest strategy of imposing identifiability constrains is based on an ordering of the emission parameters, if one exists. For instance, provided ϕ_{σ_m} are real scalar values, a labeling of the state space $\sigma'_{1:M}$ may be selected such that $\phi_{\sigma'_1} < \cdots < \phi_{\sigma'_M}$. Such an ordering is unique. In this simple case, $k^{(j)}$ may be easily identified and $\boldsymbol{\theta}_{k^{(j)}}^{(j)}$ readily found.

This strategy, of course, is problem specific and very sensitive to the parameterization of the mother distribution $\mathbb{G}_\phi$ as well as to the imposed ordering of the emission parameters ϕ. Further, it is unable to handle multivariate emission parameters or parameters that cannot be ordered on a senseful way. Below we describe an alternative strategy with higher computational cost but, although heuristic in nature, relies less on parametrizations.

For this strategy, we first need to select a reference point $\hat{\boldsymbol{\theta}}$ that we can use to compare $\boldsymbol{\theta}_k^{(j)}$ against. Subsequently, for each j, out of $\boldsymbol{\theta}_{1:K}^{(j)}$, we select $\boldsymbol{\theta}_{k^{(j)}}^{(j)}$ that yields the best comparison. The reference $\hat{\boldsymbol{\theta}}$ can be either an *ad hoc* chosen point in the space of $(\boldsymbol{s}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$ or the MCMC sample with the highest posterior value. The latter can be readily found *post hoc* among the computed MCMC values $\boldsymbol{\theta}_1^{(j)}$.

Once an appropriate reference $\hat{\boldsymbol{\theta}}$ is selected, the comparison can be based on a dissimilarity function $\mathcal{D}\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)$ that we also need to choose. For example, if $\mathcal{D}\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)$ is based on the Euclidean distance, then selection reduces to, out of $\boldsymbol{\theta}_{1:K}^{(j)}$, finding this one that belongs to the same semi-orthand with $\hat{\boldsymbol{\theta}}$.

Note 8.17: Dissimilarity function

A *dissimilarity function* $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ to every pair $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ evaluates a positive real scalar that quantifies the dissimilarity between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. For example, for two identical samples $\boldsymbol{\theta} = \boldsymbol{\theta}'$, the dissimilarity must be zero; while, for different samples $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$ the dissimilarly must be strictly positive. Solely for restoring identifiability, $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ need not be symmetric. For instance, $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ and $\mathcal{D}(\boldsymbol{\theta}', \boldsymbol{\theta})$ may attain different value.

A computationally convenient family of $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is offered by those that are additive over the dissimilarities of individual state labels

$$\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{m=1}^{M} \mathcal{E}_m\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)$$

where $\mathcal{E}_m\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)$ is a dissimilarity function that compares *only* σ_m of $\boldsymbol{\theta}$ with σ_m' of $\boldsymbol{\theta}'$.

In this case, finding the best $\boldsymbol{\theta}_k^{(j)}$ out of $\boldsymbol{\theta}_{1:K}^{(j)}$, reduces to a liner assignment problem, namely to finding the best association between the labeling $\sigma_{1:M}$ employed in $\boldsymbol{\theta}$ and the labeling $\boldsymbol{\sigma}_{1:M}'$ employed in θ'. As such, it can be solved efficiently through the Hungarian algorithm without explicitly forming each one of the K samples $\boldsymbol{\theta}_{1:K}^{(j)}$.

8.5 Dynamical variants of the Bayesian HMM

As mention earlier, with the Bayesian HMM we have flexibility that we do not have with the frequentist HMM. For example, we may consider hierarchical constructs where we place hyper-priors on $\boldsymbol{\beta}_{\sigma_m}$ and, as we see in the next section, we may develop a HMM whose state-space $\sigma_{1:M}$ may safely grow to arbitrarily large size. Consequently, such a construction may avoid the pitfalls of having to specify a particular size M to begin with, which is often a serious limitation when studying uncharacterized systems.

Before we turn to the study of uncharacterized systems, however, we focus on systems that are already characterized sufficiently well. For several such systems, properly tuning the prior on $\boldsymbol{\rho}$ and $\boldsymbol{\Pi}$ is sufficient to represent a variety of interesting dynamics. The different scenarios are endless and here we restrict only to few very important ones.

8.5.1 Modeling time scales

Due to the Markov dynamics built in the sequence of occupying states $s_{1:N}$, once a system modeled by a HMM visits a constitutive state σ_m, it remains for a random number of *additional* steps D_{σ_m} before it moves out and jumps to another constitutive state. Specifically, in section 2.4.4, we derived the distribution

$$D_{\sigma_m} | \pi_{\sigma_m \to \sigma_m} \sim \text{Geometric}(1 - \pi_{\sigma_m \to \sigma_m})$$

which depends exclusively upon the self-transition probability $\pi_{\sigma_m \to \sigma_m}$. In turn, under the prior of section 8.4.1, we immediately get $\pi_{\sigma_m \to \sigma_m} \sim \text{Beta}\left(\alpha_{\sigma_m}\beta_{\sigma_m \to \sigma_m}, \alpha_{\sigma_m}(1 - \beta_{\sigma_m \to \sigma_m})\right)$, which we may use to derive the induced prior on D_{σ_m}, which is

$$D_{\sigma_m} \sim \text{BetaNegBinomial}\left(1, \alpha_{\sigma_m}\beta_{\sigma_m \to \sigma_m}, \alpha_{\sigma_m}(1 - \beta_{\sigma_m \to \sigma_m})\right)$$

This implies that the priors applied on an HMM's transition probabilities impact upon the time scales induced. For instance, since the mean of D_{σ_m} is

$$\langle D_{\sigma_m} \rangle = \frac{\alpha_{\sigma_m} \beta_{\sigma_m \to \sigma_m}}{\alpha_{\sigma_m}(1 - \beta_{\sigma_m \to \sigma_m}) - 1}$$

we can tune α_{σ_m} and $\beta_{\sigma_m \to \sigma_m}$ so we influence dwell periods of a desired duration. For example, if a duration $\langle D_{\sigma_m} \rangle$ is specified, setting

$$\alpha_{\sigma_m} = \frac{\langle D_{\sigma_m} \rangle}{(1 - \beta_{\sigma_m \to \sigma_m})\langle D_{\sigma_m} \rangle - \beta_{\sigma_m \to \sigma_m}}$$

provides a recipe for adjusting the values of α_{σ_m} that allows for state specific time scales.

Note 8.18: The sticky HMM

One way of influencing the *same* time scale across all constitutive states in a Bayesian HMM proceeds via setting every $\alpha_{\sigma_m} = \alpha$ equal and reparametrizating $\boldsymbol{\beta}_{\sigma_m}$ as

$$\boldsymbol{\beta}_{\sigma_m} = (1 - c)\boldsymbol{B} + c\boldsymbol{D}_{\sigma_m}$$

where c is a scalar chosen between 0 and 1; $\boldsymbol{B} = [B_{\sigma_1}, \cdots, B_{\sigma_M}]$ is a probability array; while, $\boldsymbol{D}_{\sigma_m} = [D_{\sigma_m \to \sigma_1}, \cdots, D_{\sigma_m \to \sigma_M}]$ is a probability array specific to each constitutive state σ_m. The latter can be used to separate self-transitions by setting $D_{\sigma_m \to \sigma_m} = 1$ and $D_{\sigma_m \to \sigma_{m'}} = 0$. The resulting prior on $\boldsymbol{\Pi}$ consists of

$$\boldsymbol{\pi}_{\sigma_m} \sim \text{Dirichlet}_M \left(\alpha(1 - c)\boldsymbol{B} + \alpha c \boldsymbol{D}_{\sigma_m}\right), \qquad m = 1, \ldots, M.$$

With this prior, self-transitions over the entire state space are reinforced with only a limited number of hyper-parameters $\alpha, c, \boldsymbol{B}$. Because of its ability to reinforce self-transitions and so long dwells on each σ_m, this prior is termed *sticky*. Under the sticky prior, the induced dwell durations are

$$\langle D_{\sigma_m} \rangle = \frac{c + (1 - c)B_{\sigma_m}}{(1 - c)\left(1 - B_{\sigma_m}\right) - \frac{1}{\alpha}}$$

which become uniform over $\sigma_{1:M}$ by setting $B_{\sigma_m} = 1/M$.

8.5.2 Modeling equilibrium

Provided every $\beta_{\sigma_m \to \sigma_{m'}}$ is non-zero, the prior on $\boldsymbol{\Pi}$ ensures that the transition probabilities $\pi_{\sigma_m \to \sigma_{m'}}$ in a Bayesian HMM are strictly positive. In turn, this ensures that transitions between any pair of constituting states is possible in all resulting $s_{1:N}$. Therefore, a system modeled by such a HMM is ergodic, that is, it may explore the entire state-space. Such systems, if allowed to evolve for sufficiently large time may reach equilibrium.

For a dynamical system that is in equilibrium, initialization and kinetics are interrelated. In particular, the condition is that ρ_{σ_m} and $\pi_{\sigma_m \to \sigma_{m'}}$ satisfy the balance condition

$$\rho_{\sigma_m} = \sum_{\sigma_{m'}} \rho_{\sigma_{m'}} \pi_{\sigma_{m'} \to \sigma_m}, \qquad m = 1, \ldots, M$$

For a dynamical system in equilibrium, we can use this condition to express $\boldsymbol{\rho}$ in terms of $\boldsymbol{\Pi}$, which indicates that for a system in equilibrium, $\boldsymbol{\rho}$ is a dependent parameter.

Accordingly, to model a system in equilibrium we need to place priors only on $\boldsymbol{\Pi}$. For example

$$\begin{aligned}
\boldsymbol{\pi}_{\sigma_m} &\sim \text{Dirichlet}_M \left(\alpha_{\sigma_m} \boldsymbol{\beta}_{\sigma_m}\right), && m = 1, \ldots, M \\
\phi_{\sigma_m} &\sim \mathbb{H}, && m = 1, \ldots, M \\
s_1 | \boldsymbol{\Pi} &\sim \text{Categorical}_{\boldsymbol{\sigma}_{1:M}} \left(\boldsymbol{\rho}_{\boldsymbol{\Pi}}\right), \\
s_n | s_{n-1}, \boldsymbol{\Pi} &\sim \text{Categorical}_{\boldsymbol{\sigma}_{1:M}} \left(\boldsymbol{\pi}_{\sigma_m}\right), && n = 2, \ldots, N \\
w_n | s_n, \boldsymbol{\phi} &\sim \mathbb{G}_{\phi_{s_n}}, && n = 1, \ldots, N
\end{aligned}$$

in which $\boldsymbol{\rho}_{\boldsymbol{\Pi}}$ is obtained by the equilibrium condition.

Although the prior on $\boldsymbol{\Pi}$ is the same as in the HMM we saw so far, due to the implicit dependence in $\boldsymbol{\rho}_{\boldsymbol{\Pi}}$, it is not conjugate to $s_{1:N}|\boldsymbol{\Pi}$ anymore. Consequently, we cannot use the Gibbs updates of eq. (*8.45*) to obtain MCMC samples $\boldsymbol{\pi}_{\sigma_m}|s_{1:N}$. Therefore, inference is this model is possible only with the Metropolis-Hastings sampler of algorithm 8.9. Of course, in this sampler, no proposals for the initial probabilities are needed.

Note 8.19: A reversible HMM

The prior above enforces equilibrium on the HMM which is somewhat stronger than simply ensuring reversibly of the kinetics irrespective of equilibrium being reached by the time of the first measurement of not. To model a reversible dynamical system, that may not necessarily have reached equilibrium before the measurement onset, we need to consider independent priors on $\boldsymbol{\rho}$ and $\boldsymbol{\Pi}$. In such case, $\boldsymbol{\rho} \sim \text{Dirichlet}_M(\eta\boldsymbol{\zeta})$ remains an appropriate choice; however, ensuring reversible kinetics requires fundamentally different choices for $\boldsymbol{\Pi}$.

One way to ensure a reversible $\boldsymbol{\Pi}$ is to reparametrize the transition probabilities as

$$\pi_{\sigma_m\to\sigma_{m'}} = \frac{\lambda_{\sigma_m\leftrightarrow\sigma_{m'}}}{\sum_{\sigma_{m''}}\lambda_{\sigma_m\leftrightarrow\sigma_{m''}}}, \qquad m' = 1,\dots,M, \qquad m = 1,\dots,M.$$

Reversibility is ensured by requiring the new parameters to be pairwise symmetric

$$\lambda_{\sigma_m\leftrightarrow\sigma_{m'}} = \lambda_{\sigma_{m'}\leftrightarrow\sigma_m}, \qquad m' = 1,\dots,M, \qquad m = 1,\dots,M.$$

Because of symmetry, in the new parametrization, we need only $M(M+1)/2$ priors that we may choose independently. For instance

$$\lambda_{\sigma_m\leftrightarrow\sigma_{m'}} \sim \text{Gamma}\left(fE_{\sigma_m}E_{\sigma_{m'}}, 1\right), \qquad m' = 1,\dots,m, \qquad m = 1,\dots,M$$

where f and $E_{\sigma_1},\cdots,E_{\sigma_M}$ are hyper-parameters that control how tightly each constitutive state couples to the others.

The kinetic scheme defined through the symmetric prior of $\lambda_{\sigma_m\leftrightarrow\sigma_{m'}}$ is reversible with respect to the equilibrium distribution

$$\boldsymbol{\rho}_* = \left[\frac{\sum_{\sigma_m}\lambda_{\sigma_1\leftrightarrow\sigma_m}}{\sum_{\sigma_m}\sum_{\sigma_{m'}}\lambda_{\sigma_{m'}\leftrightarrow\sigma_m}},\cdots,\frac{\sum_{\sigma_m}\lambda_{\sigma_M\leftrightarrow\sigma_m}}{\sum_{\sigma_m}\sum_{\sigma_{m'}}\lambda_{\sigma_{m'}\leftrightarrow\sigma_m}}\right].$$

As a result, whenever equilibrium needs to be imposed additional to reversibility, we may proceed by reparametrizing also $\boldsymbol{\rho}$ in terms of $\lambda_{\sigma_m\leftrightarrow\sigma_{m'}}$. This is a different formulation of an equilibrium HMM than the earlier one.

8.5.3 Modeling kinetic schemes

Unlike ergodic HMM that the system modeled may move freely to and from any constituting state, some physical scenarios require modeling systems that transitions between certain states are prohibited. For example, modeling irreversible chemical reactions or photo-bleaching.

From the modeling perspective, we can take advantage of the flexibility allowed by the hyper-parameters $\beta_{\sigma_m\to\sigma_{m'}}$ to model kinetic schemes. Under the prior of section 8.4.1, a transition probability $\pi_{\sigma_m\to\sigma_m}$ is zero only whenever the corresponding $\beta_{\sigma_m\to\sigma_m}$ is zero. Essentially, to ensure that the system modeled cannot undergo some transitions or undergoes other transitions into a certain order we need to set properly the sparsity pattern of

$$\begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_M \end{array}\!\!\begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \beta_{\sigma_1\to\sigma_1} & \beta_{\sigma_1\to\sigma_2} & \cdots & \beta_{\sigma_1\to\sigma_M} \\ \beta_{\sigma_2\to\sigma_1} & \beta_{\sigma_2\to\sigma_2} & \cdots & \beta_{\sigma_2\to\sigma_M} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{\sigma_M\to\sigma_1} & \beta_{\sigma_M\to\sigma_2} & \cdots & \beta_{\sigma_M\to\sigma_M} \end{bmatrix} \end{array} = \begin{bmatrix} \boldsymbol{\beta}_{\sigma_1} \\ \boldsymbol{\beta}_{\sigma_2} \\ \vdots \\ \boldsymbol{\beta}_{\sigma_M} \end{bmatrix}$$

Example 8.5: A left-to-right HMM

For example, to model a system that returns to previous constitutive states are prohibited, we may use a left-to-right structure of the form

$$\begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \end{array} \begin{array}{c} \begin{array}{ccccc} \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \sigma_5 \end{array} \\ \begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} = \begin{bmatrix} \boldsymbol{\beta}_{\sigma_1} \\ \boldsymbol{\beta}_{\sigma_2} \\ \boldsymbol{\beta}_{\sigma_3} \\ \boldsymbol{\beta}_{\sigma_4} \\ \boldsymbol{\beta}_{\sigma_5} \end{bmatrix}$$

where we have chosen $M = 5$ for simplicity. Observed for a sufficiently long period, $N \gg 1$, a system modeled like this, eventually reaches σ_5. Since the prior imposed on $\boldsymbol{\pi}_{\sigma_5}$ is deterministic, allowing only for $\boldsymbol{\pi}_{\sigma_5} = [0, 0, 0, 0, 1]$, such a system models absorbing dynamics.

8.6 The infinite Hidden Markov Model*

In the previous section, we saw how a Bayesian HMM constructed around a fixed state-space $\sigma_{1:M}$ is highly successful in identifying the characteristics of each constituting state σ_m, for example dynamic and observation parameters represented by $\boldsymbol{\rho}, \boldsymbol{\Pi}$ and $\boldsymbol{\phi}$, respectively. These characteristics are captured in the posterior $p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$ which, for the models presented so far, inevitably depends upon the size, M, of the state-space employed.

In practice, very often we need to study dynamical systems whose state-space is uncharacterized and very often our knowledge about the system at hand does not suffice even to specify a unique M. So, despite the generality and elegance of our formulations, the posterior's dependence upon M is a serious limiting factor. Extensions of the Bayesian formulation we present in the previous section are possible that result in posterior probability distributions that are independent of M which may remain unspecified or be arbitrarily large.

In particular, by building upon the Bayesian HMM and utilizing appropriate hyper-priors, that we describe shortly, we may develop a HMM version whose state-space is infinite. Such a formulation remains appropriate and may be applied even when our primary goal is to identify the characteristics of the constituting states visited by the system while the total number of available states remains unknown.

Note 8.20: Dynamics on infinite state-spaces

With an infinite state-space, our system has access to infinite constitutive states. Specifically, each time the system departs from an occupying state s_n it has infinitely may choices σ_m to land on. Provided that the system has already visited only a finite number of them, this means that, at every transition, the systems can always explore states that will be visited for the first time. In general, such a system may be visiting an unvisited state every time it makes a transition. Although such scenarios may arise (for example, birth processes of example 2.4), most often we are interested in studying systems that sporadically revisit states. For the latter systems, the number of constitutive states visited during the time course of our measurements is drastically lower that the total number of observations.

As we see shortly, a unique feature of the infinite hidden Markov model is its ability to capture dynamics on revisiting states.

As we mentioned, the posterior $p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$ of the model in eqs. (8.36) to (8.41) depends upon M. Such dependence means that with a different number of states available, different choices of kinetic $\boldsymbol{\rho}, \boldsymbol{\Pi}$ and emission $\boldsymbol{\phi}$ parameters become more/less probable under the measurements $w_{1:N}$. For example, in the extreme case where the state-space consists of a single constitutive state, σ_1, the posterior places all its mass on self-transitions

*This is an advanced topic and could be skipped on a first reading.

$\lim_{\pi_{\sigma_1\to\sigma_1}\to 1} p(\pi_{\sigma_1\to\sigma_1}|w_{1:N}) \gg 1$; while, with a very large state-space the posterior places considerably less mass on the same self-transition $\lim_{\pi_{\sigma_1\to\sigma_1}\to 1} p(\pi_{\sigma_1\to\sigma_1}|w_{1:N}) \ll 1$.

To eliminate such dependence on M, we need to reinforce state revisiting, which can be achieved by properly selecting the priors on the initial and transition probabilities $\boldsymbol{\rho}$ and $\boldsymbol{\Pi}$. One way to do so is to consider placing a common prior among all constitutive states. In this case, η and all α_{σ_m} are equal and, for simplicity, we denote them with α. Also, $\boldsymbol{\zeta}$ and all $\boldsymbol{\beta}_{\sigma_m}$ are equal and we denote them simply with $\boldsymbol{\beta} = [\beta_{\sigma_1}, \cdots, \beta_{\sigma_M}]$. Under this common prior, constitutive states with high β_{σ_m} generally receive more transitions into them than constitutive states with low β_{σ_m}.

Of course, for an uncharacterized system we cannot identify beforehand how often the constitutive states are visited or even which of them are visited more often than the others. So, in principle, the prior $\boldsymbol{\beta}$ is unknown too and we need to estimated it in parallel with the other estimates of interest. For this reason, we place a hyper-prior on $\boldsymbol{\beta}$ and, because $\boldsymbol{\beta}$ is a probability array, the most natural choice for it is also a Dirichlet distribution

$$\boldsymbol{\beta} \sim \text{Dirichlet}_M\left(\gamma\boldsymbol{\xi}\right) \tag{8.47}$$

$$\boldsymbol{\rho}|\boldsymbol{\beta} \sim \text{Dirichlet}_M\left(\alpha\boldsymbol{\beta}\right) \tag{8.48}$$

$$\boldsymbol{\pi}_{\sigma_m}|\boldsymbol{\beta} \sim \text{Dirichlet}_M\left(\alpha\boldsymbol{\beta}\right), \qquad m = 1, \ldots, M \tag{8.49}$$

where γ is a positive scalar. As our system is uncharacterized, at this stage, because we cannot distinguish among the constitutive states, we need to ensure symmetry of $\boldsymbol{\beta}$, which we may achieve through

$$\boldsymbol{\xi} = \left[\frac{1}{M}, \cdots, \frac{1}{M}\right].$$

As anticipated, the hierarchical prior of eqs. (8.47) and (8.48), when combined with the HMM's kinetics and emissions

$$\phi_{\sigma_m} \sim \mathbb{H}, \qquad m = 1, \ldots, M \tag{8.50}$$

$$s_1|\boldsymbol{\rho} \sim \text{Categorical}_{\boldsymbol{\sigma}_{1:M}}\left(\boldsymbol{\rho}\right), \tag{8.51}$$

$$s_n|s_{n-1}, \boldsymbol{\Pi} \sim \text{Categorical}_{\boldsymbol{\sigma}_{1:M}}\left(\boldsymbol{\pi}_{\sigma_m}\right), \qquad n = 2, \ldots, N \tag{8.52}$$

$$w_n|s_n, \boldsymbol{\phi} \sim \mathbb{G}_{\phi_{s_n}}, \qquad n = 1, \ldots, N \tag{8.53}$$

results in a posterior $p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$ that converges at the limit $M \to \infty$. Consequently, as long as M is sufficiently large, the hidden Markov model above provides estimates that are independent of the particular values chosen.

Computational inference on this model can be based on appropriate modifications of the Gibbs or Metropolis-Hastings samplers of algorithms 8.7 and 8.9. The modifications for the latter are straightforward, so here we focus only on a presentation of the Gibbs sampler which targets the complete posterior $p\left(s_{1:N}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N}\right)$. For this target, only an additional step to update $\boldsymbol{\beta}$ is required in algorithm 8.7. Naively, this update needs to sample $\boldsymbol{\beta}$ from its full conditional $p\left(\boldsymbol{\beta}|s_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, w_{1:N}\right)$ which reduces to $p\left(\boldsymbol{\beta}|\boldsymbol{\rho}, \boldsymbol{\Pi}\right)$; but, because eq. (8.47) is not conjugate with eqs. (8.48) and (8.49), a Metropolis-Hastings step is necessary.

Note 8.21: iHMM

The description and the associated computational schemes we presented in this section rely on an finite approximation of the *infinite hidden Markov model* (iHMM). Formally, the latter is the model achieved at the limiting case $M = \infty$ and entails a truly infinite state-space $\sigma_{1:\infty}$. At this limit, a detailed description of the corresponding generative model is also possible; however, such description involves non-standard distributions such as the Dirichlet and Hierarchical Dirichlet process that are beyond our scope.

Additionally, it is also possible to carry our computational inference in the exact iHMM instead of relying on finite approximations. For example, it is possible to carry out MCMC sampling involving an infinite state-space by

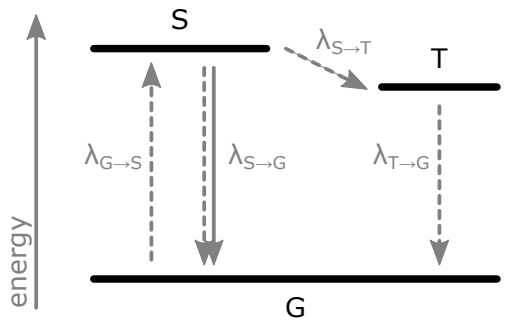


Figure 8.4: Jablonski diagram of a fluorophore possessing three energy states G, S, T. Arrows indicate Markovian transitions at the rates shown. Solid and dashed arrows distinguish between detectable and non-detectable transitions.

completing the posterior

$$p\left(s_{1:N}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\Pi} | w_{1:N}\right) = \int_{u_{1:N}} du_{1:N}\ p\left(u_{1:N}, s_{1:N}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\Pi} | w_{1:N}\right) \tag{8.54}$$

with auxiliary slice variables $u_{1:N}$. The resulting sampler, termed *beam sampler*, however, suffers from poor mixing, especially when large datasets are considered, $N \gg 1$. Instead, for such cases, the truncation at a finite, but large M, shows better performance.

8.7 A case study in fluorescence spectroscopy*

Favoring simplicity, so far we focused on problems where the observations we intent to analyze depend directly on the underlying hidden states or, as we might call them, on 1st order HMM. To illustrate that the methods we present are more general than what appears at first, we describe a case study involving dynamics in continuous time that, upon discretization, lead to 2nd order HMM. In this case study, we demonstrate how to discretize time in order to incorporate continuous time observations and how to adapt the general theory of HMM for training the resulting model.

8.7.1 Time resolved spectroscopy

An important class of experiments does not probe the state of the dynamical system of interest but rather the jumps in the system's trajectory. For instance, spectroscopic experiments held in *time-resolving mode* collect individual photons and report their detection time. Since the detected photons carry energy, which stems from the probed physical system that is typically a single molecule, they are emitted precisely when the molecule jumps across energy levels. Since the time a photon needs to reach the detector in such experiments is insignificant, the recorded photon detection times reflect the transitions between, rather than the instantaneous, states of the molecule.

In this case study, we consider a fluorescent molecule, *i.e.* a fluorophore, that has three energy states which we label with G, S, T. Respectively, these are: the ground state of the fluorophore (state with the lowest energy), the first excited singlet state (state with the highest energy), and the first excited triplet state (state with intermediate energy). These are typically depicted schematically, in increasing energy order, using a Jablonski diagram that looks like fig. 8.4.

During an experiment, while residing in G, a fluorophore absorbs energy at a random time and undergoes a transition $G \to S$. Subsequently, after residing for a short period in S, the fluorophore undergoes either an

*This is an advanced topic and could be skipped on a first reading.

$S \to G$ or an $S \to T$ transition, and once in T, the fluorophore may only undergo a $T \to G$ transition. All these are denoted with the Jablonski arrows in the diagram of fig. 8.4. Ending up at G, the fluorophore is re-excited and the same cycle repeats until the conclusion of the experiment. Physical chemistry postulates that dwells in each one of the three states are memoryless. This leads to a kinetic scheme that is fully determined by the transition rates $\lambda_{G\to S}, \lambda_{S\to G}, \lambda_{S\to T}, \lambda_{T\to G}$ which are also shown in fig. 8.4.

What makes this set-up important is the fact that the mean dwell time in the excited state S, *i.e.* the so called fluorescence lifetime, is characteristic of the fluorophore and so, once its value is pinpointed accurately, it provides information on chemical composition that might be difficult to obtain else-wise. What makes this set-up challenging to analyze though is the fact that photons are emitted and detected only whenever the fluorophore undergoes the transition $S \to G$; while, in a typical experiment, the other transitions are either non-radiative or emit photons that are not detected. The situation is even more complicated due to the fact that, even when the fluorophore undergoes $S \to G$ transitions, photons may not always be emitted or may not always be detected. Here we formulate this system and show how the general theory of HMM can be used to estimate the transition rates and eventually, through them, the fluorescence lifetime.

8.7.2 Discretization of time

For clarity, we consider an experiment that starts at time $T_{\min}$ and concludes at time $T_{\max}$. Further, we use T_k, with indices $k = 1, \dots, K$, to denote the reported photon detection times which we arrange in ascending order, *i.e.* $T_{k-1} < T_k$.

First, we need to discretize time. For this, we break the experiment's time course into a total of N hypothetical windows that are separated by the time levels

$$t_n = T_{\min} + \frac{n}{N}\left(T_{\max} - T_{\min}\right), \qquad n = 0, 1, \dots, N.$$

These time levels define N windows which we index successively by $n = 1, \dots, N$. Specifically, our n^{th} window spans the time interval between t_{n-1} and t_n.

8.7.3 Formulation of the dynamics

Following the notation we first introduce in chapter 2, section 2.3, we denote with $\mathcal{S}(t)$ the occupying state at time t of our fluorophore. Due to memorylessness, the trajectory $\mathcal{S}(\cdot)$ is a Markov jump process with state-space G, S, T and its transition rate matrix is given by

$$\mathbf{\Lambda} = \begin{bmatrix} 0 & \lambda_{G\to S} & \lambda_{G\to T} \\ \lambda_{S\to G} & 0 & \lambda_{S\to T} \\ \lambda_{T\to G} & \lambda_{T\to S} & 0 \end{bmatrix}.$$

Our end goal is to estimate the unknown entries of $\mathbf{\Lambda}$. To do so, we do not need the full trajectory $\mathcal{S}(\cdot)$. Instead, we focus on the occupying states only at the time levels t_n which are already sufficient to link $\mathbf{\Lambda}$ with our measurements. Accordingly, for each time level we consider the corresponding occupying state

$$s_n = \mathcal{S}(t_n), \qquad n = 0, 1, \dots, N.$$

Because the underlying trajectory is a Markov jump process, we can easily deduce the transition rules of our dynamical model

$$s_n | s_{n-1} \sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}_{s_{n-1}}\right), \qquad n = 1, \dots, N.$$

According to eq. (2.14), the transition probabilities stem from the rows of the propagator

$$\mathbf{\Pi} = \begin{bmatrix} \boldsymbol{\pi}_G \\ \boldsymbol{\pi}_S \\ \boldsymbol{\pi}_T \end{bmatrix} = \begin{bmatrix} \pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} \\ \pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} \\ \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \end{bmatrix} = \boldsymbol{Q}^{t_{n-1}\to t_n} = \mathbf{exp}\left(\frac{T_{\max} - T_{\min}}{N}\boldsymbol{G}\right) \tag{8.55}$$

that corresponds to the generator $\boldsymbol{G}$ of the rate matrix $\boldsymbol{\Lambda}$. As with every dynamical system we saw so far, the kinetic model does not specify the initial conditions. Consequently, we need to model the initialization rule separately

$$s_0 \sim \text{Categorical}_{G,S,T}(\boldsymbol{\rho})$$

with appropriate initial probabilities $\boldsymbol{\rho} = [\rho_G, \rho_S, \rho_T]$ that may be the related or underrated to $\boldsymbol{\Lambda}$ depending upon the specifics of the experiment.

8.7.4 Formulation of the measurements

The most convenient way to model the photon detection times is to consider a set of observation variables $w_{1:N}$, where each one of our windows is associated with its own w_n. We encode the photon detection times $T_{1:K}$, by having $w_n = 1$ provided at least one photon is detected and $w_n = 0$ provided no photon is detected during our n^{th} window.

Note 8.22: Observations

If we use N_k to denote the window that encodes the k^{th} photon detection time, T_k, we see that

$$N_k = \left\lceil N \frac{T_k - T_{\min}}{T_{\max} - T_{\min}} \right\rceil, \qquad k = 1, \ldots, K,$$

where $\lceil x \rceil$ is the ceiling function, *i.e.* the smallest index that is larger than x.

When we attempt to model our photon detections with a low N, our windows may be large and misleadingly, some of them, may engulf more than one photon detections. However, as N grows large and our windows shrink, the photon detections times $T_{1:K}$ are encoded in different windows and these windows are well separated. Specifically, for *sufficiently* large N, our observation variables $w_{1:N}$ follow the pattern

$$\underbrace{0, \cdots, 0}_{\substack{\text{windows} \\ 1:N_1-1}}, \overbrace{1}^{T_1}, \underbrace{0, \cdots, 0}_{\substack{\text{windows} \\ N_1+1:N_2-1}}, \overbrace{1}^{T_2}, \underbrace{0, \cdots, 0}_{\substack{\text{windows} \\ N_2+1:N_3-1}}, \overbrace{1}^{T_3}, 0 \cdots \qquad \cdots 0, \overbrace{1}^{T_K}, \underbrace{0, \cdots, 0}_{\substack{\text{windows} \\ N_K+1:N}}.$$

Due to this pattern, our observation sequence $w_{1:N}$ contains *no successive* windows with $w_n = 1$; in contrast, it contains multiple successive windows with $w_n = 0$.

Under the variables $w_{1:N}$, it is straightforward to model our assessment rules by

$$w_n | s_{n-1}, s_n \sim \text{Bernoulli}\left(\beta_{s_{n-1} \to s_n}\right), \qquad n = 1, \ldots, N,$$

and, because we have 9 possible pairs $s_{n-1} \to s_n$, we need to specify in total 9 different Bernoulli weights. To a good approximation, these are given by

$$\begin{aligned}
&\beta_{G\to G} \approx 0, && \beta_{G\to S} \approx 0, && \beta_{G\to T} \approx 0,\\
&\beta_{S\to G} \approx \eta, && \beta_{S\to S} \approx 0, && \beta_{S\to T} \approx 0,\\
&\beta_{T\to G} \approx 0, && \beta_{T\to S} \approx 0, && \beta_{T\to T} \approx 0,
\end{aligned}$$

where η is the fraction of detectable transitions $S \to G$ to total transitions $S \to G$.

Our approximations on $\beta_{s_{n-1}\to s_n}$ improve and eventually become exact at the limit $N \to \infty$ at which our windows become so thin that they can accommodate no more than one transition each. For this reason, our end goal is to devise a training method that supports this limit. Put differently, our strategy is to derive a set of training equations on which we can obtain the limit $N \to \infty$ formally.

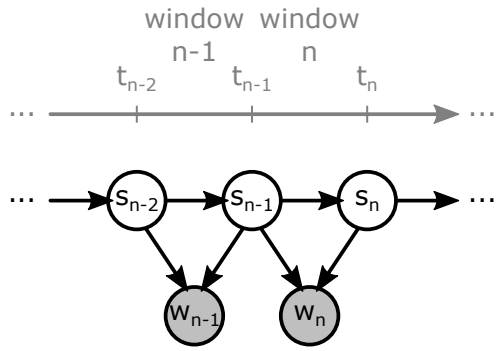


Figure 8.5: A HMM for fluoresce spectroscopy represents time resolved measurements, $T_{1:K}$, by observation variables w_n that are linked to the occupying states s_n of the underling fluorophore. In contrast to the observation variables $w_{1:N}$ that are measured in an experiment, the occupying states $s_{0:N}$ remain hidden.

8.7.5 Modeling overview

In summary, the model of time resolved fluorescence spectroscopy we developed so far takes the form

$$\begin{aligned} s_0 &\sim \text{Categorical}_{G,S,T}(\boldsymbol{\rho}) \\ s_n|s_{n-1} &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}_{s_{n-1}}\right), && n = 1,\dots,N, \\ w_n|s_{n-1},s_n &\sim \text{Bernoulli}\left(\beta_{s_{n-1}\to s_n}\right), && n = 1,\dots,N, \end{aligned}$$

and is depicted graphically in fig. 8.5. An immediate challenge that we face with this model is that each observation variable $w_n|s_{n-1},s_n$ depends on *two,* rather than one, hidden states. Because of this minor, but otherwise important difference, we cannot apply any of the basic algorithms of section 8.3.

8.7.6 Reformulation

To continue, we need to reformulate our model in such a way that is becomes similar to the HMM we devised earlier. Namely, we need to transform it such that each observation is associated with *only one* hidden state.

One way to achieve a transformation is to consider a time period τ that is positive, $\tau > 0$, but sufficient small, $\tau < (T_{\max} - T_{\min})/N$. With the aid of τ, we can consider two additional occupying states per time level

$$u_n = \mathcal{S}\left(t_{n-1} + \frac{\tau}{2}\right), \qquad v_n = \mathcal{S}\left(t_n - \frac{\tau}{2}\right), \qquad n = 1,\dots,N.$$

From the new states, u_n occurs near the very beginning and v_n occurs near the very end of their respective window. Even with the introduction of these states, due to memorylessness, we can still represent exactly the dynamics of our system

$$\begin{aligned} u_n|s_{n-1} &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\psi}'_{s_{n-1}}\right), && n = 1,\dots,N, \\ v_n|u_n &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}'_{u_n}\right), && n = 1,\dots,N, \\ s_n|v_n &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\psi}'_{v_n}\right), && n = 1,\dots,N. \end{aligned}$$

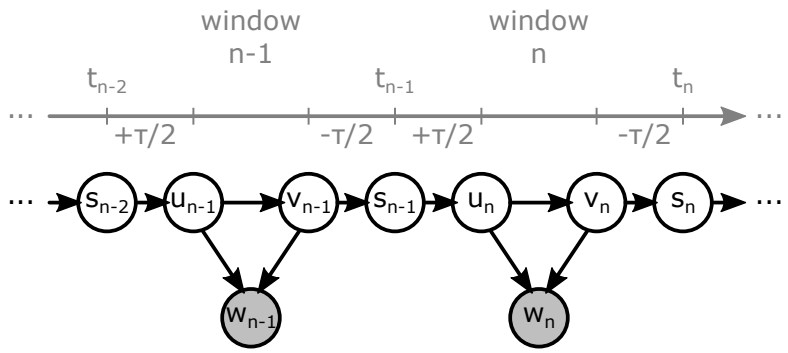


Figure 8.6: A HMM, augmented with additional hidden states $u_{1:N}, v_{1:N}$, is used to decouple successive occupying states $s_{0:N}$ from their respective observations $w_{1:N}$.

The new transition probabilities are obtained through the rows of the propagators

$$\boldsymbol{\Psi}' = \begin{bmatrix} \boldsymbol{\psi}'_G \\ \boldsymbol{\psi}'_S \\ \boldsymbol{\psi}'_T \end{bmatrix} = \begin{bmatrix} \psi'_{G\to G} & \psi'_{G\to S} & \psi'_{G\to T} \\ \psi'_{S\to G} & \psi'_{S\to S} & \psi'_{S\to T} \\ \psi'_{T\to G} & \psi'_{T\to S} & \psi'_{T\to T} \end{bmatrix} = \boldsymbol{Q}^{t_{n-1}\to t_{n-1}+\frac{\tau}{2}} = \boldsymbol{Q}^{t_n-\frac{\tau}{2}\to t_n} = \mathbf{exp}\left(\frac{\tau}{2}\boldsymbol{G}\right),$$

$$\boldsymbol{\Pi}' = \begin{bmatrix} \boldsymbol{\pi}'_G \\ \boldsymbol{\pi}'_S \\ \boldsymbol{\pi}'_T \end{bmatrix} = \begin{bmatrix} \pi'_{G\to G} & \pi'_{G\to S} & \pi'_{G\to T} \\ \pi'_{S\to G} & \pi'_{S\to S} & \pi'_{S\to T} \\ \pi'_{T\to G} & \pi'_{T\to S} & \pi'_{T\to T} \end{bmatrix} = \boldsymbol{Q}^{t_{n-1}+\frac{\tau}{2}\to t_n-\frac{\tau}{2}} = \mathbf{exp}\left(\left(\frac{T_{\max}-T_{\min}}{N}-\tau\right)\boldsymbol{G}\right).$$

Taking advantage of the new states, and provided τ is sufficiently small, we can introduce another approximation to the observations

$$\beta_{s_{n-1}\to s_n} \approx \beta_{u_n\to v_n}, \qquad\qquad n = 1,\ldots,N.$$

This approximation becomes exact at the limit $\tau \to 0^+$ at which u_n and v_n essentially merge with s_{n-1} and s_n, respectively. Of course, because $\tau < (T_{\max}-T_{\min})/N$, this limiting condition does not introduce further restrictions in our formulation since it is already fulfilled under $N\to\infty$.

Gathering everything together, our reformulated model consists of the equations

$$\begin{aligned}
s_0 &\sim \mathsf{Categorical}_{G,S,T}(\boldsymbol{\rho}), \\
u_n|s_{n-1} &\sim \mathsf{Categorical}_{G,S,T}\left(\boldsymbol{\psi}'_{s_{n-1}}\right), && n = 1,\ldots,N, \\
v_n|u_n &\sim \mathsf{Categorical}_{G,S,T}\left(\boldsymbol{\pi}'_{u_n}\right), && n = 1,\ldots,N, \\
s_n|v_n &\sim \mathsf{Categorical}_{G,S,T}\left(\boldsymbol{\psi}'_{v_n}\right), && n = 1,\ldots,N, \\
w_n|u_n,v_n &\sim \mathsf{Bernoulli}\left(\beta_{u_n\to v_n}\right), && n = 1,\ldots,N,
\end{aligned}$$

which we depict graphically in fig. 8.6. Now, because the states $s_{0:N}$ are not directly associated with the observations anymore, we can afford to discard them by marginalization, which leads us to an equivalent, but somewhat simpler, version

$$\begin{aligned}
u_1 &\sim \mathsf{Categorical}_{G,S,T}(\boldsymbol{\rho}'), \\
v_1|u_1 &\sim \mathsf{Categorical}_{G,S,T}\left(\boldsymbol{\pi}'_{u_1}\right), \\
u_n|v_{n-1} &\sim \mathsf{Categorical}_{G,S,T}\left(\boldsymbol{\psi}''_{v_{n-1}}\right), && n = 2,\ldots,N,
\end{aligned}$$

$$v_n|u_n \sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}'_{u_n}\right), \qquad n = 2,\dots,N,$$
$$w_n|u_n, v_n \sim \text{Bernoulli}\left(\beta_{u_n \to v_n}\right), \qquad n = 1,\dots,N,$$

which we depict graphically in the left panel of fig. 8.7. Marginalization implies that, in this model, the initial probabilities are given by

$$\boldsymbol{\rho}' = \begin{bmatrix}\rho'_G & \rho'_S & \rho'_T\end{bmatrix} = \boldsymbol{\rho}\boldsymbol{\Phi}' = \boldsymbol{\rho}\,\mathbf{exp}\left(\frac{\tau}{2}\boldsymbol{G}\right)$$

and the transition probabilities by the rows of

$$\boldsymbol{\Psi}'' = \begin{bmatrix}\boldsymbol{\psi}''_G \\ \boldsymbol{\psi}''_S \\ \boldsymbol{\psi}''_T\end{bmatrix} = \begin{bmatrix}\psi''_{G\to G} & \psi''_{G\to S} & \psi''_{G\to T} \\ \psi''_{S\to G} & \psi''_{S\to S} & \psi''_{S\to T} \\ \psi''_{T\to G} & \psi''_{T\to S} & \psi''_{T\to T}\end{bmatrix} = \boldsymbol{\Psi}'\boldsymbol{\Psi}' = \mathbf{exp}\left(\tau\boldsymbol{G}\right).$$

Note 8.23: HMM order reduction

The last version of our model represents a conventional HMM as introduced in section 8.2. To make the correspondence more clear, we consider super-states $\xi_n = (u_n, v_n)$, which we depict graphically in the right panel of fig. 8.7, and rewrite the model in the equivalent form

$$\xi_1 \sim \text{Categorical}_{\chi_{1:9}}(\boldsymbol{r}),$$
$$\xi_n|\xi_{n-1} \sim \text{Categorical}_{\chi_{1:9}}\left(\boldsymbol{P}_{\xi_{n-1}}\right), \qquad n = 2,\dots,N,$$
$$w_n|\xi_n \sim \text{Bernoulli}\left(\beta_{\xi_n}\right), \qquad n = 1,\dots,N.$$

The initial, $\boldsymbol{r}$, and transition, $\boldsymbol{P}_\xi$, probabilities are determined according to $\boldsymbol{\rho}', \boldsymbol{\Pi}', \boldsymbol{\Psi}''$. In particular, these are

$$\begin{aligned} r_{\xi_1} = p(\xi_1) &= p(u_1, v_1) \\ &= p(v_1|u_1)p(u_1) = \pi'_{u_1\to v_1}\rho'_{u_1}, \end{aligned}$$
$$\begin{aligned} P_{\xi_{n-1}\to\xi_n} = p(\xi_n|\xi_{n-1}) &= p(u_n, v_n|u_{n-1}, v_{n-1}) \\ &= p(v_n|u_n, u_{n-1}, v_{n-1})p(u_n|u_{n-1}, v_{n-1}) \\ &= p(v_n|u_n)p(u_n|v_{n-1}) = \pi'_{u_n\to v_n}\psi''_{v_{n-1}\to u_n}. \end{aligned}$$

Because each super-state is formed by a pair of G, S, T, our new state-space consists of

$$\chi_1 = GG, \quad \chi_2 = GS, \quad \chi_3 = GT,$$
$$\chi_4 = SG, \quad \chi_5 = SS, \quad \chi_6 = ST,$$
$$\chi_7 = TG, \quad \chi_8 = TS, \quad \chi_9 = TT,$$

and, because each constitutive super-state is *derived* from G, S, T, similar to example 2.10, we follow the common convention and order $\chi_{1:9}$ lexicographically.

8.7.7 Computational training

So far, we managed to re-formulate our problem and make it amenable to a similar training strategy as the conventional HMM of sections 8.3 and 8.4. In its final version, the unknown parameters are still those of the initial problem, namely the entries of $\boldsymbol{\Lambda}$ and potentially $\boldsymbol{\rho}, \eta$. The likelihood of our model, formally given by $L = p(w_{1:N}|\boldsymbol{\Lambda}, \boldsymbol{\rho}, \eta)$, can be computed according to eq. (8.8) by completion with the terminal states

$$L = p\left(w_{1:N}|\boldsymbol{\Lambda}, \boldsymbol{\rho}, \eta\right) = \sum_{u_N, v_N} p\left(w_{1:N}, u_N, v_N|\boldsymbol{\Lambda}, \boldsymbol{\rho}, \eta\right) = \sum_{u_N, v_N} \mathcal{A}_N(u_N, v_N).$$

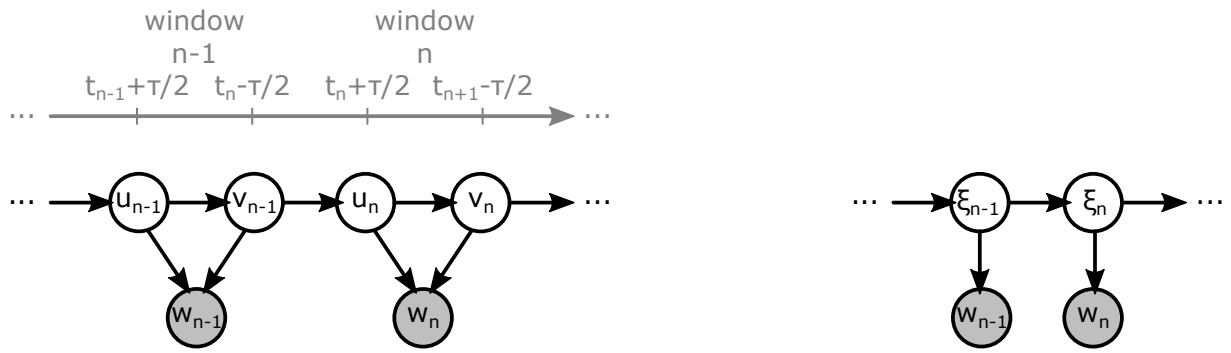


Figure 8.7: Left: A modified HMM with two decoupled occupying states per observation. Right: equivalent HMM with one occupying state per observation.

In turn, the terms $\mathcal{A}_N(u_N, v_N)$ can be computed by forward filtering. Nevertheless, because N needs to be large, so our approximate observation representation holds, naive filtering with algorithm 8.1 is impractical. Additionally, even if we were able to perform the filtering recursion in algorithm 8.1 for excessively large N, directly training our model suffers from the approximations induced by having a non-zero τ and a finite N. Below, we show how to eliminate such approximations altogether and how to derive a tractable version of the filtering algorithm that carries over the limit $N \to \infty$.

Limit $\tau \to 0^+$

Because all of our propagators depends continuously on τ, we can formally apply the limit $\tau \to 0^+$. Specifically, note 2.10 implies that

$$\mathbf{exp}\left(\frac{\tau}{2}\boldsymbol{G}\right) \to \boldsymbol{I}, \qquad \mathbf{exp}\left(\tau \boldsymbol{G}\right) \to \boldsymbol{I}, \qquad \mathbf{exp}\left(\left(\frac{T_{\max} - T_{\min}}{N} - \tau\right)\boldsymbol{G}\right) \to \boldsymbol{\Pi}.$$

Therefore, we can safely replace our model with the limiting one

$$\begin{aligned}
u_1 &\sim \text{Categorical}_{G,S,T}(\boldsymbol{\rho}), \\
v_1|u_1 &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}_{u_1}\right), \\
u_n|v_{n-1} &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{I}_{v_{n-1}}\right), && n = 2, \ldots, N, \\
v_n|u_n &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}_{u_n}\right), && n = 2, \ldots, N, \\
w_n|u_n, v_n &\sim \text{Bernoulli}\left(\beta_{u_n \to v_n}\right), && n = 1, \ldots, N,
\end{aligned}$$

and effectively relax any approximation mediated by τ.

Marginal likelihood

Having relaxed the dependence on τ, we now show how to apply forward filtering, *i.e.* algorithm 8.1. To make our calculations more transparent, we adopt the super-state formalism, $\xi_n = (u_n, v_n)$, of note 8.23 and show how to compute recursively the forward terms which, in this case, read $\mathcal{A}_n(u_n, v_n) = \mathcal{A}_n(\xi_n)$. Further, to maintain the notation to a minimum, we follow note 8.8 and gather our forward terms in row arrays

$$\mathbb{A}_n = \begin{bmatrix} \mathcal{A}_n(\chi_1) & \mathcal{A}_n(\chi_2) & \mathcal{A}_n(\chi_3) & \mathcal{A}_n(\chi_4) & \mathcal{A}_n(\chi_5) & \mathcal{A}_n(\chi_6) & \mathcal{A}_n(\chi_7) & \mathcal{A}_n(\chi_8) & \mathcal{A}_n(\chi_9) \end{bmatrix}.$$

With this convention, the computation of the (marginal) likelihood reduces to

$$L = \mathbb{A}_N \boldsymbol{\Sigma}, \qquad \boldsymbol{\Sigma} = \boldsymbol{\sigma} \otimes \boldsymbol{\sigma}, \qquad \boldsymbol{\sigma} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Note 8.24: Vectorization

According to note 8.23, the model in section 8.7.7, leads to the tabulations

$$\boldsymbol{r} = \begin{bmatrix} \rho_G \pi_{G\to G} & \rho_G \pi_{G\to S} & \rho_G \pi_{G\to T} & \rho_S \pi_{S\to G} & \rho_S \pi_{S\to S} & \rho_S \pi_{S\to T} & \rho_T \pi_{T\to G} & \rho_T \pi_{T\to S} & \rho_T \pi_{T\to T} \end{bmatrix}$$

$$\boldsymbol{P} = \begin{bmatrix} \pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \\ \pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \\ \pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \end{bmatrix}$$

Adopting array operations, both are vectorized

$$\boldsymbol{r} = \left(\boldsymbol{\rho} \otimes \boldsymbol{\sigma}^t\right) \odot \left(\boldsymbol{a}_G^t \boldsymbol{\Pi} \boldsymbol{B}_G + \boldsymbol{a}_S^t \boldsymbol{\Pi} \boldsymbol{B}_S + \boldsymbol{a}_T^t \boldsymbol{\Pi} \boldsymbol{B}_T\right),$$

$$\boldsymbol{P} = \left(\boldsymbol{\sigma} \otimes \boldsymbol{I} \otimes \boldsymbol{\sigma}^t\right) \odot \left(\boldsymbol{A}_G^t \boldsymbol{\Pi} \boldsymbol{B}_G + \boldsymbol{A}_S^t \boldsymbol{\Pi} \boldsymbol{B}_S + \boldsymbol{A}_T^t \boldsymbol{\Pi} \boldsymbol{B}_T\right),$$

where $\otimes, \odot$ denote the Kronecker and Hadamard product, respectively, and the auxiliary arrays are

$$\boldsymbol{a}_G = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{A}_G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B}_G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{a}_S = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{A}_S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B}_S = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{a}_T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{A}_T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{B}_T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

From eq. (*8.10*), we see that the filtering updates follow the recursion

$$\mathcal{A}_n(\xi_n) = \sum_{\xi_{n-1}} \left(\text{Bernoulli}(w_n; \beta_{\xi_n}) P_{\xi_{n-1}\to\xi_n}\right) \mathcal{A}_{n-1}(\xi_{n-1}), \qquad n = 2, \ldots, N$$

which we can vectorize in

$$\mathbb{A}_n = \boldsymbol{P}_{w_n} \mathbb{A}_{n-1}, \qquad n = 2, \ldots, N.$$

Note 8.25: Vectorization

The matrices, $\boldsymbol{P}_0$ and $\boldsymbol{P}_1$, required in the filtering updates are tabulated in

$$\boldsymbol{P}_0 = \begin{bmatrix}
\pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \zeta_0\pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \\
\pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \zeta_0\pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \\
\pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \zeta_0\pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T}
\end{bmatrix}$$

$$\boldsymbol{P}_1 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \zeta_1\pi_{S\to G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \zeta_1\pi_{S\to G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \zeta_1\pi_{S\to G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

with $\zeta_0 = 1-\eta$ and $\zeta_1 = \eta$. Similar to $\boldsymbol{P}$, these are vectorized by

$$\boldsymbol{P}_0 = \left(\boldsymbol{\sigma}\otimes\boldsymbol{I}\otimes\boldsymbol{\sigma}^t\right)\odot\left(\boldsymbol{A}_G^t\boldsymbol{\Pi}_0\boldsymbol{B}_G + \boldsymbol{A}_S^t\boldsymbol{\Pi}_0\boldsymbol{B}_S + \boldsymbol{A}_T^t\boldsymbol{\Pi}_0\boldsymbol{B}_T\right),$$
$$\boldsymbol{P}_1 = \left(\boldsymbol{\sigma}\otimes\boldsymbol{I}\otimes\boldsymbol{\sigma}^t\right)\odot\left(\boldsymbol{A}_G^t\boldsymbol{\Pi}_1\boldsymbol{B}_G + \boldsymbol{A}_S^t\boldsymbol{\Pi}_1\boldsymbol{B}_S + \boldsymbol{A}_T^t\boldsymbol{\Pi}_1\boldsymbol{B}_T\right).$$

In $\boldsymbol{P}_0$ and $\boldsymbol{P}_1$, we use $\boldsymbol{\Pi}_0$ and $\boldsymbol{\Pi}_1$ to discriminate between detection-less and detection-full pseudo-propagators

$$\boldsymbol{\Pi}_0 = \boldsymbol{Z}_0\odot\boldsymbol{\Pi}, \qquad \boldsymbol{\Pi}_1 = \boldsymbol{Z}_1\odot\boldsymbol{\Pi},$$

where, with the masks, $\boldsymbol{Z}_0$ and $\boldsymbol{Z}_1$, we encode detection-less and detection-full transitions

$$\boldsymbol{Z}_0 = \begin{bmatrix} 1 & 1 & 1 \\ \zeta_0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad \boldsymbol{Z}_1 = \begin{bmatrix} 0 & 0 & 0 \\ \zeta_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

As $\boldsymbol{Z}_0$ and $\boldsymbol{Z}_1$ encode our observation rules, the pseudo-propagators are related by $\boldsymbol{\Pi} = \boldsymbol{\Pi}_0 + \boldsymbol{\Pi}_1$. Additionally, because photons are emitted only when our system *jumps* across constitutive states, the diagonal entires in $\boldsymbol{Z}_0$ are all 1; in contrast, the diagonal entries in $\boldsymbol{Z}_1$ are all 0.

Finally, according to eq. (*8.11*), the filter is initialized with $\mathcal{A}_1(\xi_1) = \text{Bernoulli}(w_1;\beta_{\xi_1})r_{\xi_1}$ which, in vectorized form reads

$$\mathbb{A}_1 = \left(\boldsymbol{\rho}\otimes\boldsymbol{\sigma}^t\right)\odot\left(\boldsymbol{a}_G^t\boldsymbol{\Pi}_{w_1}\boldsymbol{B}_G + \boldsymbol{a}_S^t\boldsymbol{\Pi}_{w_1}\boldsymbol{B}_S + \boldsymbol{a}_T^t\boldsymbol{\Pi}_{w_1}\boldsymbol{B}_T\right).$$

Note 8.26: Vectorization

With the aid of two operators

$$\mathbb{L}\left(\boldsymbol{C}\right) = \left(\boldsymbol{\rho}\boldsymbol{C}\right)\otimes\boldsymbol{\sigma}^t, \qquad \mathbb{D}\left(\boldsymbol{C}\right) = \boldsymbol{a}_G^t\boldsymbol{C}\boldsymbol{B}_G + \boldsymbol{a}_S^t\boldsymbol{C}\boldsymbol{B}_S + \boldsymbol{a}_T^t\boldsymbol{C}\boldsymbol{B}_T$$

defined over the 3×3 matrices $\boldsymbol{C}$; the initial forward term takes a much simpler form

$$\mathbb{A}_1 = \mathbb{L}\left(\boldsymbol{I}\right)\odot\mathbb{D}\left(\boldsymbol{\Pi}_{w_1}\right).$$

By induction, we can now show that the forward variables, $\mathbb{A}_{1:N}$, satisfy an important relationship

$$\mathbb{A}_n = \mathbb{L}\left(\boldsymbol{I}\boldsymbol{\Pi}_{w_1}\cdots\boldsymbol{\Pi}_{w_{n-1}}\right)\odot\mathbb{D}\left(\boldsymbol{\Pi}_{w_n}\right), \qquad n=1,\dots,N.$$

Accordingly, the (marginal) likelihood is given by

$$L = \left[\mathbb{L}\left(\boldsymbol{\Pi}_{w_1}\cdots\boldsymbol{\Pi}_{w_{N-1}}\right)\odot\mathbb{D}\left(\boldsymbol{\Pi}_{w_N}\right)\right]\boldsymbol{\Sigma} = \mathbb{L}\left(\boldsymbol{\Pi}_{w_1}\cdots\boldsymbol{\Pi}_{w_{N-1}}\right)\left[\mathbb{D}\left(\boldsymbol{\Pi}_{w_N}\right)\right]^t = \boldsymbol{\rho}\,\boldsymbol{\Pi}_{w_1}\cdots\boldsymbol{\Pi}_{w_N}\boldsymbol{\sigma}.$$

Limit $N\to\infty$

According to note 8.22, the product of the pseudo-propagators in our likelihood has the form

$$\begin{aligned}\boldsymbol{\Pi}_{w_1}\cdots\boldsymbol{\Pi}_{w_N} &= \overbrace{\boldsymbol{\Pi}_0\cdots\boldsymbol{\Pi}_0}^{\substack{\text{windows}\\1:N_1-1}}\boldsymbol{\Pi}_1\overbrace{\boldsymbol{\Pi}_0\cdots\boldsymbol{\Pi}_0}^{\substack{\text{windows}\\N_1+1:N_2-1}}\boldsymbol{\Pi}_1\boldsymbol{\Pi}_0\cdots \qquad \cdots\boldsymbol{\Pi}_0\boldsymbol{\Pi}_1\overbrace{\boldsymbol{\Pi}_0\cdots\boldsymbol{\Pi}_0}^{\substack{\text{windows}\\N_K+1:N}}\\ &= \boldsymbol{\Pi}_0^{N_1-1}\boldsymbol{\Pi}_1\boldsymbol{\Pi}_0^{N_2-N_1-1}\boldsymbol{\Pi}_1\cdots \qquad \cdots\boldsymbol{\Pi}_0^{N_K-N_{K-1}-1}\boldsymbol{\Pi}_1\boldsymbol{\Pi}_0^{N-N_K}.\end{aligned}$$

Note 8.27: Asymptotics

Considering the limit $N\to\infty$, from eq. (8.55), we see

$$\boldsymbol{\Pi} = \boldsymbol{I} + \frac{T_{\max}-T_{\min}}{N}\boldsymbol{G} + \mathcal{O}\left(\frac{1}{N^2}\right),$$

which we may also use on the pseudo-propagators. Specifically, because $\boldsymbol{\Pi}_0 = \boldsymbol{\Pi}\odot\boldsymbol{Z}_0$ and $\boldsymbol{\Pi}_1 = \boldsymbol{\Pi}\odot\boldsymbol{Z}_1$, we readily derive

$$\begin{aligned}\boldsymbol{\Pi}_0 &= \boldsymbol{I} + \frac{T_{\max}-T_{\min}}{N}\boldsymbol{G}_0 + \mathcal{O}\left(\frac{1}{N^2}\right) = \mathbf{exp}\left(\frac{T_{\max}-T_{\min}}{N}\boldsymbol{G}_0\right) + \mathcal{O}\left(\frac{1}{N^2}\right)\\ \boldsymbol{\Pi}_1 &= \frac{T_{\max}-T_{\min}}{N}\boldsymbol{G}_1 + \mathcal{O}\left(\frac{1}{N^2}\right)\end{aligned}$$

where $\boldsymbol{G}_0 = \boldsymbol{G}\odot\boldsymbol{Z}_0$ and $\boldsymbol{G}_1 = \boldsymbol{G}\odot\boldsymbol{Z}_1$.

Additionally, according to note 8.22, we have

$$N_k\frac{T_{\max}-T_{\min}}{N} = \frac{T_{\max}-T_{\min}}{N} + \mathcal{O}\left(\frac{1}{N^2}\right), \qquad k=1,\dots,K.$$

Combining everything together, we obtain the asymptotic expression of our likelihood

$$L = \left(\frac{T_{\max}-T_{\min}}{N}\right)^K\ell + \mathcal{O}\left(\frac{1}{N^{K+1}}\right). \tag{8.56}$$

where ℓ is *independent* of N. Specifically, ℓ is given by

$$\begin{aligned}\ell = \boldsymbol{\rho}\,\mathbf{exp}\left(g_0\boldsymbol{G}_0\right)\boldsymbol{G}_1\mathbf{exp}\left(g_1\boldsymbol{G}_0\right)\boldsymbol{G}_1\cdots\\ \cdots\boldsymbol{G}_1\mathbf{exp}\left(g_{K-1}\boldsymbol{G}_0\right)\boldsymbol{G}_1\mathbf{exp}\left(g_K\boldsymbol{G}_0\right)\boldsymbol{\sigma}.\end{aligned}$$

As can be seen, ℓ depends only on $\boldsymbol{\Lambda},\boldsymbol{\rho},\eta$ and the successive time lags

$$g_0 = T_1 - T_{\min}, \qquad g_1 = T_2 - T_1, \qquad \cdots \qquad g_{K-1} = T_K - T_{K-1}, \qquad g_K = T_{\max} - T_K.$$

8.7.8 Bayesian considerations

From eq. (8.56), it becomes clear that the unknown parameters in our formulation enter the likelihood of the model in a complicated way that renders it pointless to seek training through the Baum-Welch method of section 8.3.3. Similarly, unless under very special circumstances, Bayesian training with conjugate priors, like those in section 8.4.1, is also pointless.

A viable training strategy, however, is through a Metropolis-Hastings MCMC scheme where, under non-conjugate prior assignments, proposals are drawn and subsequently accepted or rejected according to the (marginal) posterior. Because this strategy is quite general, here we consider a wider problem, where the unknown parameters may include not only entries of the transition rate matrix $\mathbf{\Lambda}$, but also initial probabilities $\boldsymbol{\rho}$ and observation parameter η.

For clarity, we gather the unknown parameters in θ and, to stress out the dependence on them, we denote with $\ell(\theta)$ the product in eq. (8.56). With this formalism, our priors, which need to be specified, are encoded in $p(\theta)$ and our likelihood is given, only asymptotically, by

$$p(w_{1:N}|\theta) = \left(\frac{T_{\max} - T_{\min}}{N}\right)^K \ell(\theta) + \mathcal{O}\left(\frac{1}{N^{K+1}}\right).$$

As in section 5.2.1, using an appropriate Metropolis-Hastings proposal $q(\theta^{\text{prop}}|\theta^{\text{old}})$, we arrive at the acceptance ratio, eq. (5.10), of the form

$$A_N\left(\theta^{\text{prop}}|\theta^{\text{old}}\right) = \frac{p\left(w_{1:N}|\theta^{\text{prop}}\right)}{p\left(w_{1:N}|\theta^{\text{old}}\right)} \frac{p\left(\theta^{\text{prop}}\right)}{p\left(\theta^{\text{old}}\right)} \frac{q\left(\theta^{\text{old}}|\theta^{\text{prop}}\right)}{q\left(\theta^{\text{prop}}|\theta^{\text{old}}\right)}.$$

For any finite choice of N, this ratio is intractable. However, the limiting case $N \to \infty$ leads to

$$A_\infty\left(\theta^{\text{prop}}|\theta^{\text{old}}\right) = \frac{\ell\left(\theta^{\text{prop}}\right)}{\ell\left(\theta^{\text{old}}\right)} \frac{p\left(\theta^{\text{prop}}\right)}{p\left(\theta^{\text{old}}\right)} \frac{q\left(\theta^{\text{old}}|\theta^{\text{prop}}\right)}{q\left(\theta^{\text{prop}}|\theta^{\text{old}}\right)}$$

which we can easily evaluate numerically.

Example 8.6: Bayesian fluorescence spectroscopy

In the most general case, the unknowns in a typical problem of interest in fluorescence spectroscopy may include: all transition rates $\lambda_{G\to S}, \lambda_{S\to G}, \lambda_{S\to T}, \lambda_{T\to G}$, all initial probabilities ρ_G, ρ_S, ρ_T, as well as η. Convenient choices for the priors are as following

$$\lambda_{G\to S} \sim \text{Gamma}\left(2, \frac{\lambda_{\text{ref}}}{2}\right), \qquad \lambda_{S\to G} \sim \text{Gamma}\left(2, \frac{\lambda_{\text{ref}}}{2}\right),$$
$$\lambda_{S\to T} \sim \text{Gamma}\left(2, \frac{\lambda_{\text{ref}}}{2}\right), \qquad \lambda_{T\to G} \sim \text{Gamma}\left(2, \frac{\lambda_{\text{ref}}}{2}\right),$$
$$\boldsymbol{\rho} \sim \text{Dirichlet}_3\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), \qquad \eta \sim \text{Beta}\,(1,1)$$

In these priors, the hyper-parameters may be adjusted to incorporate prior confidence on certain values and λ_{ref} can be used to set an appropriate time scale.

For numerical stability, it is preferable if unit-less priors are used

$$\tilde{\lambda}_{G\to S} \sim \text{Gamma}\left(2, \frac{1}{2}\right), \qquad \tilde{\lambda}_{S\to G} \sim \text{Gamma}\left(2, \frac{1}{2}\right)$$
$$\tilde{\lambda}_{S\to T} \sim \text{Gamma}\left(2, \frac{1}{2}\right), \qquad \tilde{\lambda}_{T\to G} \sim \text{Gamma}\left(2, \frac{1}{2}\right)$$

and a time scale is implemented through eq. (8.56) cast in the form

$$\ell = \boldsymbol{\rho}\,\mathbf{exp}\left(\tilde{g}_0\tilde{\boldsymbol{G}}_0\right)\tilde{\boldsymbol{G}}_1\mathbf{exp}\left(\tilde{g}_1\tilde{\boldsymbol{G}}_0\right)\tilde{\boldsymbol{G}}_1\cdots$$

$$\cdots \tilde{\boldsymbol{G}}_1 \mathbf{exp}\left(\tilde{g}_{K-1}\tilde{\boldsymbol{G}}_0\right) \tilde{\boldsymbol{G}}_1 \mathbf{exp}\left(\tilde{g}_K \tilde{\boldsymbol{G}}_0\right) \boldsymbol{\sigma}.$$

with $\tilde{g}_k = g_k \lambda_{\mathrm{ref}}$. Numerical stability can be further increased if the fastest time scale is separated and evaluated analytically. In particular, if $\tilde{\lambda}_{\mathrm{fast}}$ denotes the fastest rate, $\tilde{\boldsymbol{G}}_0$ can be replaced by $\tilde{\boldsymbol{\Gamma}}_0 - \tilde{\lambda}_{\mathrm{fast}} \boldsymbol{I}$. This way, ℓ results to

$$\ell = e^{-\tilde{\lambda}_{\mathrm{fast}} \lambda_{\mathrm{ref}} (T_{\max} - T_{\min})} \boldsymbol{\rho} \mathbf{exp}\left(\tilde{g}_0 \tilde{\boldsymbol{\Gamma}}_0\right) \tilde{\boldsymbol{G}}_1 \mathbf{exp}\left(\tilde{g}_1 \tilde{\boldsymbol{\Gamma}}_0\right) \tilde{\boldsymbol{G}}_1 \cdots \\ \cdots \tilde{\boldsymbol{G}}_1 \mathbf{exp}\left(\tilde{g}_{K-1} \tilde{\boldsymbol{\Gamma}}_0\right) \tilde{\boldsymbol{G}}_1 \mathbf{exp}\left(\tilde{g}_K \tilde{\boldsymbol{\Gamma}}_0\right) \boldsymbol{\sigma}.$$

8.8 Exercise problems

Exercise 8.1: EM for Poisson HMM

Describe and implement an EM training strategy on a HMM with Poisson emissions. For concreteness, consider the model

$$\begin{aligned} s_1 | \boldsymbol{\rho} &\sim \mathsf{Categorical}_{\sigma_{1:M}}(\boldsymbol{\rho}) \\ s_n | s_{n-1}, \boldsymbol{\Pi} &\sim \mathsf{Categorical}_{\sigma_{1:M}}(\boldsymbol{\pi}_{\sigma_m}), && n = 2, \ldots, N \\ w_n | s_n, \boldsymbol{\phi} &\sim \mathsf{Poisson}(\phi_{s_n}), && n = 1, \ldots, N \end{aligned}$$

and follow algorithm 8.4.

Exercise 8.2: A Bayesian HMM for FRET measurements

Use the FRET context of example 8.2 to:

1. Set up a Bayesian HMM for the analysis of measurements w_n^D, w_n^A.
2. Describe an MCMC sampling scheme for the posterior of part 1.
3. Implement the sampling scheme of part 2.
4. Consider the approximations of example 8.2 and set up a Bayesian HMM for the analysis of apparent FRET efficiencies ϵ_n.
5. Describe the modifications in your MCMC scheme of part 2 required to sample the posterior of part 4.

Chapter 9

State-space models

In this chapter we continue our study of *time dependent measurements*. Specifically, we concentrate on continuous space systems that evolve in discrete space. Due to computational limitations we mostly consider linear Gaussian systems and present a detailed description of the *Kalman theory*.

9.1 State-space models

In many cases of practical interest, the driving dynamics behind a physical system are fully or partially unobserved. For example, as we have seen in Add references to previous chapters, on a fundamental level the evolution of a system may depend on microscopic or even quantum events, interactions among numerous components, unknown or uncharacterized physics and so on. Obviously, such factors cannot be quantified directly. Nevertheless, often they need to be accounted for in order to facilitate the interpretation of observations either at the same or higher levels.

When dynamical variables remain unassessed, specialized models that are data-friendly but also physically accurate need to be applied. We have already seen in ref HMM chapter how hidden Markov models (HMM) can be used in conjunction with empirical data for this task. However, for several physical systems, the hidden variables may not be limited to only discrete values, such as "on/off" states or more abstractly $\sigma_{1:M}$, that we have assumed in the formulation of HMM. Instead, for most physical systems, dynamical variables may attain a continuum of values, for example positions, intensities, concentrations, etc. Obviously, such random variables cannot be sampled through $\text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\pi}_{\sigma_{1:M}})$ distributions that are fundamental in the development and training of HMM. For such applications, *state-space models*, that we describe in this chapter, provide a flexible extension that preserves the statistical structure of HMM but may accommodate continuous random variables.

In particular, state-space models allow us to model a sequence of observations $\boldsymbol{y}_n$, obtained at times t_n, as being driven by an array of (possibly unobserved) state variables $\boldsymbol{x}_n$ that, in turn, are driven by a stochastic process. The variables $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ may be scalar or array-valued and the number of time levels n may be either finite, i.e. $n = 1, ..., N$ for some appropriate N, or even infinite, i.e. $n = 1, 2, \ldots$.

A state-space model, in a fairly general form, is summarized in the following pair of causal relations that we have already encountered in previous chapters Ref observation process and such. In particular, these can be viewed as

$$\boldsymbol{x}_n | \boldsymbol{x}_{n-1} \sim G^n_{\boldsymbol{x}_{n-1}} \tag{9.1}$$

$$\boldsymbol{y}_n | \boldsymbol{x}_n \sim F^n_{\boldsymbol{x}_n} \tag{9.2}$$

?? describes the dynamics of the unobserved state $\boldsymbol{x}_n$, which are influenced by $\boldsymbol{x}_{n-1}$, and **??** describes the generation of the measurements $\boldsymbol{y}_n$ which are influenced by $\boldsymbol{x}_n$. In these equations, $G^n_{\boldsymbol{x}_{n-1}}$ and $F^n_{\boldsymbol{x}_n}$ are the probability densities of the associated random variables which, in pedantic notation, are $\boldsymbol{X}_n | \boldsymbol{x}_{n-1}$ and $\boldsymbol{Y}_n | \boldsymbol{x}_n$. In particular, $G^n_{\boldsymbol{x}_{n-1}}$ describes a stochastic relation between successive states which is assumed Markovian and may account for internal processes; while $F^n_{\boldsymbol{x}_n}$ describes a stochastic relation between measurements and states that may account for external processes such as observation errors or noise. Both, dynamics $G^n_{\boldsymbol{x}_n}$ and observations $F^n_{\boldsymbol{x}_n}$, despite of depending on the states, they may also change over time.

In the general form given above, the state-space formulation places no restrictions on the densities $G^n_{\boldsymbol{x}_{n-1}}$ and $F^n_{\boldsymbol{x}_n}$ besides that these are densities of continuous random variables. For most practical applications, however, it is sufficient (or even desirable) to consider two special classes that we list next.

First, the standard approach is to model $G^n_{\boldsymbol{x}_{n-1}}$ and $F^n_{\boldsymbol{x}_n}$ as containing additive Gaussian noises. In this case, the densities can be cast as

$$G^n_{\boldsymbol{x}_{n-1}} = \text{Normal}\,(g_n(\boldsymbol{x}_{n-1}), V_n) \tag{9.3}$$

$$F^n_{\boldsymbol{x}_n} = \text{Normal}\,(f_n(\boldsymbol{x}_{n-1}), U_n) \tag{9.4}$$

for some, generally non-linear, functions $g_n(\boldsymbol{x})$ and $f_n(\boldsymbol{x})$ that describe the system in the absence of any noise. In this class of state-space models, V_n and U_n are the covariances of the dynamics and measurements, respectively, that together with $g_n(\boldsymbol{x})$ and $f_n(\boldsymbol{x})$ are problem specific.

Second, for a limited number of applications it is sufficient to model the functions $g_n(\boldsymbol{x})$ and $f_n(\boldsymbol{x})$ as linear. For instance

$$g_n(\boldsymbol{x}) = B_n\boldsymbol{x} + b_n \tag{9.5}$$

$$f_n(\boldsymbol{x}) = A_n\boldsymbol{x} + a_n \tag{9.6}$$

for some matrices B_n, A_n and vectors b_n, a_n of appropriate dimensions that are consistent with $\boldsymbol{x}_n, \boldsymbol{y}_n$ as well as U_n, V_n.

Example 9.1

Consider the case of Brownian motion in 3D that is observed under noisy measurements. As usual, consider $\boldsymbol{x}_n$ to be the true position, respectively, at time t_n. Further, consider a Stokesian particle and let the temperature be externally controlled.

Describe 2 cases: constant temperature and time varying temperature.

Describe 2 special cases of dynamics: non-linear forces and linear forces.

Describe 2 special cases of time levels: equidistant and non-equidistant.

Describe 3 special cases of observations: $\boldsymbol{y}_n$ gaussian (such as pre-localized super-resolution microscopy); $\boldsymbol{w}_n$ with PSF and gaussian intensity, $\boldsymbol{w}_n$ with PSF and poisson intensity. For these cases consider only thin pixels.

State-space models can be formulated much more general than the classes we highlight here and we provide a limited overview of alternatives in section 9.3. In any case, no matter which class is more appropriate to work with, there are typically three broad questions to be asked:

1. Given observations $\boldsymbol{y}_{1:N}$ up to time t_N, how do we estimate states $\boldsymbol{x}_n$ at times *after* t_N?
2. Given observations $\boldsymbol{y}_{1:N}$ up to time t_N, how do we estimate states $\boldsymbol{x}_n$ at times *before* t_N?
3. Given observations $\boldsymbol{y}_{1:N}$ up to time t_N, how do we simulate a state sequence $\boldsymbol{x}_{1:N}$?

As we show in the next section, these questions can be answered in terms of the probability densities $p\,(\boldsymbol{x}_n|\boldsymbol{y}_{1:N})$ for either $n < N$ or $n = N$ or $n > N$.

The efficient computation of these densities is the main focus of the theory in this chapter. Nevertheless, to construct each density $p\,(\boldsymbol{x}_n|\boldsymbol{y}_{1:N})$ it is necessary first to cast the random variables $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ in the Bayesian framework. For this, it is sufficient and convenient to consider a state $\boldsymbol{x}_0$ corresponding to time t_0, that is not associated with any observation, and place a prior probability distribution only on $\boldsymbol{x}_0$. Since the remaining $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ have already specified distributions by **????** the resulting model is in full Bayesian form.

In summary, our entire Bayesian representation from now on is

$$\boldsymbol{x}_0 \sim H \tag{9.7}$$

$$\boldsymbol{x}_n|\boldsymbol{x}_{n-1} \sim G^n_{\boldsymbol{x}_{n-1}}, \qquad n = 1, 2, \ldots \tag{9.8}$$

$$\boldsymbol{y}_n|\boldsymbol{x}_n \sim F^n_{\boldsymbol{x}_n}, \qquad n = 1, 2, \ldots \tag{9.9}$$

where the form of the prior H, the dynamics $G^n_{\boldsymbol{x}_{n-1}}$ and the measurements $F^n_{\boldsymbol{x}_n}$, are generally problem specific.

Note 9.1: The multivariate normal

A box with the multivariate normal as in Little p 190

9.2 Filtering, smoothing, and simulation in state-space models

9.2.1 Kalman theory for linear Gaussian models

In the simplest case where (i) the prior is Gaussian, (ii) noises are additive and Gaussian, and (iii) dynamics and measurements are linear, the state-space representation is

$$\boldsymbol{x}_0 \sim \text{Normal}\left(c, W\right) \tag{9.10}$$

$$\boldsymbol{x}_n|\boldsymbol{x}_{n-1} \sim \text{Normal}\left(B_n\boldsymbol{x}_{n-1} + b_n, V_n\right), \qquad n = 2, 3, \ldots \tag{9.11}$$

$$\boldsymbol{y}_n|\boldsymbol{x}_n \sim \text{Normal}\left(A_n\boldsymbol{x}_n + a_n, U_n\right), \qquad n = 1, 2, \ldots \tag{9.12}$$

The great advantage of this model is that, as we demonstrate below, one can compute pretty much every probability distribution of interest analytically.

The underlying theory has been studied extensively and parallels to a large degree the theory of HMM. Below we provide a brief derivation of key results.

Kalman filter

As with HMM, it is convenient to start with the computation of filter densities $p\left(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}\right)$; that is, estimates of the state $\boldsymbol{x}_n$ provided observations $\boldsymbol{y}_{1:n}$. For example, each filter density can be computed using Bays' rule $p\left(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}\right) = p\left(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{y}_{1:n-1}\right) p\left(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1}\right) / p\left(\boldsymbol{y}_{1:n}\right)$ which, for Markovian causal relations, simplifies to

$$p\left(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}\right) = \frac{p\left(\boldsymbol{y}_n|\boldsymbol{x}_n\right) p\left(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1}\right)}{p\left(\boldsymbol{y}_{1:n}\right)} \tag{9.13}$$

The necessary computations can be carried out in two stages: a *prediction stage* where the density $p\left(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1}\right)$ is computed first and a *correction stage* where the product $p\left(\boldsymbol{y}_n|\boldsymbol{x}_n\right) p\left(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1}\right)$ is computed next. Once both stages are completed, the evidence term $p\left(\boldsymbol{y}_{1:n}\right)$ can be recovered by normalization, although in most applications its explicit computation is unnecessary.

In particular, at $n = 1$ the prediction stage relies on

$$p\left(\boldsymbol{x}_1\right) = \int_{\boldsymbol{x}_0} d\boldsymbol{x}_0 p\left(\boldsymbol{x}_1, \boldsymbol{x}_0\right) \tag{9.14}$$

$$= \int_{\boldsymbol{x}_0} d\boldsymbol{x}_0 p\left(\boldsymbol{x}_1|\boldsymbol{x}_0\right) p\left(\boldsymbol{x}_0\right) \tag{9.15}$$

$$= \int_{\boldsymbol{x}_0} d\boldsymbol{x}_0 \text{Normal}\left(\boldsymbol{x}_n; B_1\boldsymbol{x}_0 + b_1, V_1\right) \text{Normal}\left(\boldsymbol{x}_0; c, W\right) \tag{9.16}$$

$$\propto \text{Normal}\left(\boldsymbol{x}_1; r_{1|0}, R_{1|0}\right) \tag{9.17}$$

where $r_{1|0} = B_1 c + b_1$ and $R_{1|0} = B_1 W_1 B_1^T + V_1$; while, the correction stage relies on

$$p(\boldsymbol{y}_1|\boldsymbol{x}_1)p(\boldsymbol{x}_1) \propto \text{Normal}\left(\boldsymbol{y}_1; A_1\boldsymbol{x}_1 + a_1, U_1\right) \text{Normal}\left(\boldsymbol{x}_1; r_{1|0}, R_{1|0}\right) \tag{9.18}$$

$$\propto \text{Normal}\left(\boldsymbol{x}_1; r_{1|1}, R_{1|1}\right) \tag{9.19}$$

where $r_{1|1}$ =??? and $R_{1|1}$ =???.

A nice homework problem is to have the students verify the expressions for $r_{1|0}, R_{1|0}$ and $r_{1|1}, R_{1|1}$. Mathematically, this problem is only basic linear algebra on multivariate gaussians. I think it is a good idea to have

the students carry out those calculations so they realize (1) how filtering works and (2) why approximations are almost always necessary.

Similarly, at $n = 2$ the prediction stage relies on

$$p(\boldsymbol{x}_2|\boldsymbol{y}_1) = \int_{\boldsymbol{x}_1} d\boldsymbol{x}_1 p(\boldsymbol{x}_2, \boldsymbol{x}_1|\boldsymbol{y}_1) \tag{9.20}$$

$$= \int_{\boldsymbol{x}_1} d\boldsymbol{x}_1 p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{y}_1) p(\boldsymbol{x}_1|\boldsymbol{y}_1) \tag{9.21}$$

$$= \int_{\boldsymbol{x}_1} d\boldsymbol{x}_1 p(\boldsymbol{x}_2|\boldsymbol{x}_1) p(\boldsymbol{x}_1|\boldsymbol{y}_1) \tag{9.22}$$

$$= \int_{\boldsymbol{x}_1} d\boldsymbol{x}_1 \text{Normal}(\boldsymbol{x}_2; B_2\boldsymbol{x}_1 + b_2, V_2) \text{Normal}(\boldsymbol{x}_1; r_{1|1}, R_{1|1}) \tag{9.23}$$

$$\propto \text{Normal}(\boldsymbol{x}_2; r_{2|1}, R_{2|1}) \tag{9.24}$$

where $r_{2|1} = B_2 r_{1|1} + b_2$ and $R_{2|1} = B_2 R_{1|1} B_2^T + V_2$; while, the correction stage relies on

$$p(\boldsymbol{y}_2|\boldsymbol{x}_2)p(\boldsymbol{x}_2|\boldsymbol{y}_1) \propto \text{Normal}(\boldsymbol{y}_2; A_2\boldsymbol{x}_2 + a_2, U_2) \text{Normal}(\boldsymbol{x}_2; r_{2|1}, R_{2|1}) \tag{9.25}$$

$$\propto \text{Normal}(\boldsymbol{x}_2; r_{2|2}, R_{2|2}) \tag{9.26}$$

where $r_{2|2}$ =??? and $R_{2|2}$ =???.

For subsequent time levels n, prediction and correction stages are very similar. As can be already seen, due to specific structure of the model, the filters for any n remain Gaussian

$$p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}) = \text{Normal}(\boldsymbol{x}_n; r_{n|n}, R_{n|n}), \qquad n = 1, 2, \ldots \tag{9.27}$$

Starting from the prior $r_{0|0} = c$ and $R_{0|0} = W$ and computing recursively, the filter computation is described compactly in algorithm 9.1.

Algorithm 9.1: Kalman filter

1. Initialization

$$r_{0|0} = c, \qquad R_{0|0} = W \tag{9.28}$$

2. For $n = 1, \ldots, N$
   (a) Prediction

$$r_{n|n-1} = B_n r_{n-1|n-1} + b_n, \tag{9.29}$$

$$R_{n|n-1} = B_n R_{n-1|n-1} B_n^T + V_n \tag{9.30}$$

   (b) Correction

$$r_{n|n} = ???, \qquad R_{n|n} = ??? \tag{9.31}$$

Obviously, our choice of subscripts $n|n$ to distinguish the characteristics (i.e. mean and covariance) of the prior and the filters is not arbitrary. In fact, when working with temporally structured data, it is custom to use $n|N$ to highlight that information on time level n is requested provided information up to time level N.

By their definition, filters summarize information about a time level provided information up to that time level too and so $n|n$ is the appropriate notation. However, in principle information may be requested for a time level that is different from time level of the last available observation. For example, we have already encountered predictions $n|n-1$, where state estimates are requested one time level ahead of the last available observation, and below we will encounter other, more complicated, situations.

Kalman forecaster

Given a finite sequence of observations $\boldsymbol{y}_{1:N}$, once the filter densities $p(\boldsymbol{x}_n|\boldsymbol{y}_{1:N})$ are computed, we can estimate future states even multiple time levels ahead. In fact, forecasting k time levels ahead, that is forecasting states at time level $n = N + k$, is almost identical to the prediction stage we have already seen.

For instance, forecasting one step ahead is based on

$$\begin{aligned} p\left(\boldsymbol{x}_{N+1}|\boldsymbol{y}_{1:N}\right) &= \int_{\boldsymbol{x}_N} d\boldsymbol{x}_N p\left(\boldsymbol{x}_{N+1}, \boldsymbol{x}_N|\boldsymbol{y}_{1:N}\right) && (9.32)\\ &= \int_{\boldsymbol{x}_N} d\boldsymbol{x}_N p\left(\boldsymbol{x}_{N+1}|\boldsymbol{x}_N\right) p\left(\boldsymbol{x}_N|\boldsymbol{y}_{1:N}\right) && (9.33)\\ &= \int_{\boldsymbol{x}_N} d\boldsymbol{x}_N \mathsf{Normal}\left(\boldsymbol{x}_{N+1}; B_{N+1}\boldsymbol{x}_N + b_{N+1}, V_{N+1}\right) \mathsf{Normal}\left(\boldsymbol{x}_N; r_{N|N}, R_{N|N}\right) && (9.34)\\ &= \mathsf{Normal}\left(\boldsymbol{x}_{N+1}; r_{N+1|N}, R_{N+1|N}\right) && (9.35) \end{aligned}$$

where $r_{N+1|N} = B_{N+1}r_{N|N} + b_{N+1}$ and $R_{N+1|N} = B_{N+1}R_{N|N}B_{N+1}^T + V_{N+1}$.

Similarly, forecasting two steps ahead is based on

$$\begin{aligned} p\left(\boldsymbol{x}_{N+2}|\boldsymbol{y}_{1:N}\right) &= \int_{\boldsymbol{x}_{N+1}} d\boldsymbol{x}_{N+1} p\left(\boldsymbol{x}_{N+2}, \boldsymbol{x}_{N+1}|\boldsymbol{y}_{1:N}\right) && (9.36)\\ &= \int_{\boldsymbol{x}_{N+1}} d\boldsymbol{x}_{N+1} p\left(\boldsymbol{x}_{N+2}|\boldsymbol{x}_{N+1}\right) p\left(\boldsymbol{x}_{N+1}|\boldsymbol{y}_{1:N}\right) && (9.37)\\ &= \int_{\boldsymbol{x}_{N+1}} d\boldsymbol{x}_{N+1} \mathsf{Normal}\left(\boldsymbol{x}_{N+2}; B_{N+2}\boldsymbol{x}_{N+1} + b_{N+2}, V_{N+2}\right) \mathsf{Normal}\left(\boldsymbol{x}_{N+1}; r_{N+1|N}, R_{N+1|N}\right) && (9.38)\\ &= \mathsf{Normal}\left(\boldsymbol{x}_{N+2}; r_{N+2|N}, R_{N+2|N}\right) && (9.39) \end{aligned}$$

where $r_{N+2|N} = B_{N+2}r_{N+1|N} + b_{N+2}$ and $R_{N+2|N} = B_{N+2}R_{N+1|N}B_{N+2}^T + V_{N+2}$.

Forecasting for additional steps is very similar. As can be seen, the forecaster k steps ahead remain Gaussian

$$p(\boldsymbol{x}_{N+1}|\boldsymbol{y}_{1:N}) = \mathsf{Normal}\left(\boldsymbol{x}_{N+k}; r_{N+k|N}, R_{N+k|N}\right), \qquad k = 1, 2, \ldots \tag{9.40}$$

Starting from the terminal filter prior $r_{N|N}$ and $R_{N|N}$ and computing recursively, the forecaster computation is described compactly in algorithm 9.2.

Algorithm 9.2: Kalman forecaster

1. Initialize by computing the terminal filter $r_{N|N}$ and $R_{N|N}$ according to algorithm 9.1
2. For $k = 1, 2, \ldots$ update the forecaster recursively by

$$r_{N+k|N} = B_{N+k}r_{N+k-1|N} + b_{N+k} \tag{9.41}$$
$$R_{N+k|N} = B_{N+k}R_{N+k-1|N}B_{N+k}^T + V_{N+k} \tag{9.42}$$

Kalman smoother

Given a finite sequence of observations $\boldsymbol{y}_{1:N}$, we can also estimate states at any time level before N. As we can now use every available observation, state estimates for $n < N$ are tighter that those provided by the corresponding filters since they are based on $\boldsymbol{x}_n|\boldsymbol{y}_{1:N}$ instead of $\boldsymbol{x}_n|\boldsymbol{y}_{1:n}$.

At any $n < N$, smoothed estimates stem from the factorization

$$p(\boldsymbol{x}_n|\boldsymbol{y}_{1:N}) = \frac{p(\boldsymbol{y}_{1:N}, \boldsymbol{x}_n)}{p(\boldsymbol{y}_{1:N})} \tag{9.43}$$

$$= \frac{p(\boldsymbol{y}_{1:N}|\boldsymbol{x}_n)p(\boldsymbol{x}_n)}{p(\boldsymbol{y}_{1:N})} \tag{9.44}$$

$$= \frac{p(\boldsymbol{y}_{1:n}|\boldsymbol{x}_n)p(\boldsymbol{y}_{n+1:N}|\boldsymbol{x}_n)p(\boldsymbol{x}_n)}{p(\boldsymbol{y}_{1:N})} \tag{9.45}$$

$$= \frac{p(\boldsymbol{y}_{1:n}, \boldsymbol{x}_n)p(\boldsymbol{y}_{n+1:N}|\boldsymbol{x}_n)}{p(\boldsymbol{y}_{1:N})} \tag{9.46}$$

$$= \frac{p(\boldsymbol{y}_{1:n})}{p(\boldsymbol{y}_{1:N})} p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})p(\boldsymbol{y}_{n+1:N}|\boldsymbol{x}_n) \tag{9.47}$$

which relies on the filter $p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})$ as well on a smoother $p(\boldsymbol{y}_{n+1:N}|\boldsymbol{x}_n)$. We have already encountered an efficient algorithm for computing the filters, for example algorithm 9.1. Computing the smoothers is similar.

For example, at $n = N - 1$ the smoother is based on

$$p(\boldsymbol{y}_N|\boldsymbol{x}_{N-1}) = \int_{\boldsymbol{x}_N} d\boldsymbol{x}_N p(\boldsymbol{y}_N, \boldsymbol{x}_N|\boldsymbol{x}_{N-1}) \tag{9.48}$$

$$= \int_{\boldsymbol{x}_N} d\boldsymbol{x}_N p(\boldsymbol{y}_N|\boldsymbol{x}_N)p(\boldsymbol{x}_N|\boldsymbol{x}_{N-1}) \tag{9.49}$$

$$= \int_{\boldsymbol{x}_N} d\boldsymbol{x}_N \text{Normal}\left(\boldsymbol{y}_N; A_N\boldsymbol{x}_N + a_N, U_N\right) \text{Normal}\left(\boldsymbol{x}_N; B_N\boldsymbol{x}_{N-1} + b_N, V_N\right) \tag{9.50}$$

$$= C_{N-1}\text{Normal}\left(\boldsymbol{x}_{N-1}; ??, ??\right) \tag{9.51}$$

where ???, ??? and C_{N-1} is a constant that does not depend on $\boldsymbol{x}_{N-1}$.

Similarly, at $n = N - 2$ the smoother is based on

$$p(\boldsymbol{y}_{N-1:N}|\boldsymbol{x}_{N-2}) = \int_{\boldsymbol{x}_{N-1}} d\boldsymbol{x}_{N-1} p(\boldsymbol{y}_{N-1:N}, \boldsymbol{x}_{N-1}|\boldsymbol{x}_{N-2}) \tag{9.52}$$

$$= \int_{\boldsymbol{x}_{N-1}} d\boldsymbol{x}_{N-1} p(\boldsymbol{y}_{N-1:N}|\boldsymbol{x}_{N-1})p(\boldsymbol{x}_{N-1}|\boldsymbol{x}_{N-2}) \tag{9.53}$$

$$= \int_{\boldsymbol{x}_{N-1}} d\boldsymbol{x}_{N-1} p(\boldsymbol{y}_N|\boldsymbol{x}_{N-1})p(\boldsymbol{y}_{N-1}|\boldsymbol{x}_{N-1})p(\boldsymbol{x}_{N-1}|\boldsymbol{x}_{N-2}) \tag{9.54}$$

$$= C_{N-1} \int_{\boldsymbol{x}_{N-1}} d\boldsymbol{x}_{N-1} \text{Normal}\left(\boldsymbol{x}_{N-1}; ??, ??\right) \text{Normal}\left(\boldsymbol{y}_{N-1}; A_{N-1}\boldsymbol{x}_{N-1} + a_{N-1}, U_{N-1}\right) \text{Normal}\left(\boldsymbol{x}_{N-1}; B_{N-1}\right.$$

(9.55)

$$= C_{N-2}\text{Normal}\left(\boldsymbol{x}_{N-2}; ??, ??\right) \tag{9.56}$$

where ???, ??? and C_{N-2} is a constant that does not depend on $\boldsymbol{x}_{N-1:N}$.

Computing smoothers for earlier steps is very similar. As can be seen, the smoother at any time level n retains a Gaussian form

$$p(\boldsymbol{y}_{n+1:N}|\boldsymbol{x}_n) = C_n\text{Normal}\left(\boldsymbol{x}_n; ???, ???\right), \qquad n = 1, 2, \ldots, N-1 \tag{9.57}$$

where none of the constants C_n depends on $\boldsymbol{x}_{n:N}$.

Once every filter $p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})$ and every smoother $p(\boldsymbol{y}_{n+1:N}, \boldsymbol{x}_n)$ area available, the smoothed estimates re-

sulting from eq. (9.58) are given by

$$p(\boldsymbol{x}_n|\boldsymbol{y}_{1:N}) = \frac{p(\boldsymbol{y}_{1:n})}{p(\boldsymbol{y}_{1:N})} p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}) p(\boldsymbol{y}_{n+1:N}|\boldsymbol{x}_n) \tag{9.58}$$

$$= \frac{p(\boldsymbol{y}_{1:n})}{p(\boldsymbol{y}_{1:N})} \text{Normal}\left(\boldsymbol{x}_n; r_{n|n}, R_{n|n}\right) C_n \text{Normal}\left(\boldsymbol{x}_n; ???, ???\right) \tag{9.59}$$

$$= \text{Normal}\left(\boldsymbol{x}_n; r_{n|N}, R_{n|N}\right) \tag{9.60}$$

where $r_{n|N} =???$ and $R_{n|N} =???$.

Unlike the filters, according to our definitions the smoothers are not properly normalized and therefore they do not represent probability densities. For example, this is the reason we need to introduce the constants C_n in eq. (9.57). Nevertheless, such constants C_n, need not be computed in eq. (9.58) since they can be recovered by normalization of eq. (9.58). In particular, these are

$$C_n = \frac{p(\boldsymbol{y}_{1:N})}{p(\boldsymbol{y}_{1:n})} \tag{9.61}$$

Kalman simulator

When there are no available observations, simulating (i.e. sampling) Overall comment: it is generally a good idea to distinguish throughout the text between sampling and simulating a state trajectory $\boldsymbol{x}_{1:N}$, for any finite N is straightforward based on the generative model of eqs. (9.10) to (9.12). Specifically, under no observations, simulated trajectories follow $p(\boldsymbol{x}_{1:N})$ which, due to the causality of the model, factorizes as

$$p(\boldsymbol{x}_{1:N}) = \int_{\boldsymbol{x}_0} d\boldsymbol{x}_0 p(\boldsymbol{x}_0, \boldsymbol{x}_{1:N}) \tag{9.62}$$

$$= \int_{\boldsymbol{x}_0} d\boldsymbol{x}_0 p(\boldsymbol{x}_0) p(\boldsymbol{x}_1|\boldsymbol{x}_0) p(\boldsymbol{x}_2|\boldsymbol{x}_1) \cdots p(\boldsymbol{x}_N|\boldsymbol{x}_{N-1}) \tag{9.63}$$

$$= \int_{\boldsymbol{x}_0} d\boldsymbol{x}_0 H(\boldsymbol{x}_0) G^1_{\boldsymbol{x}_1}(\boldsymbol{x}_0) G^2_{\boldsymbol{x}_2}(\boldsymbol{x}_1) \cdots G^N_{\boldsymbol{x}_N}(\boldsymbol{x}_{N-1}) \tag{9.64}$$

which already suggests a sequential procedure that starts by sampling a state $\boldsymbol{x}_0$ from the prior, and subsequently utilizing dynamics to sample following states.

In contrast, *given a sequence of observations* $\boldsymbol{y}_{1:N}$, simulating a state trajectories $\boldsymbol{x}_{1:N}$ is less straightforward since trajectories need to follow the posterior $p(\boldsymbol{x}_{1:N}|\boldsymbol{y}_{1:N})$ which factorizes in a more complex form. Similar to the HMM, the factorization is

$$p(\boldsymbol{x}_{1:N}|\boldsymbol{y}_{1:N}) = p(\boldsymbol{x}_1|\boldsymbol{x}_{2:N}, \boldsymbol{y}_{1:N}) p(\boldsymbol{x}_2|\boldsymbol{x}_{3:N}, \boldsymbol{y}_{1:N}) \cdots p(\boldsymbol{x}_{N-1}|\boldsymbol{x}_N, \boldsymbol{y}_{1:N}) p(\boldsymbol{x}_N|\boldsymbol{y}_{1:N}) \tag{9.65}$$

$$= p(\boldsymbol{x}_1|\boldsymbol{x}_2, \boldsymbol{y}_1) p(\boldsymbol{x}_2|\boldsymbol{x}_3, \boldsymbol{y}_{1:2}) \cdots p(\boldsymbol{x}_{N-1}|\boldsymbol{x}_N, \boldsymbol{y}_{1:N-1}) p(\boldsymbol{x}_N|\boldsymbol{y}_{1:N}) \tag{9.66}$$

$$\propto p(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{y}_1) p(\boldsymbol{x}_2, \boldsymbol{x}_3|\boldsymbol{y}_{1:2}) \cdots p(\boldsymbol{x}_{N-1}, \boldsymbol{x}_N|\boldsymbol{y}_{1:N-1}) p(\boldsymbol{x}_N|\boldsymbol{y}_{1:N}) \tag{9.67}$$

$$= p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{y}_1) p(\boldsymbol{x}_1|\boldsymbol{y}_1) p(\boldsymbol{x}_3|\boldsymbol{x}_2, \boldsymbol{y}_{1:2}) p(\boldsymbol{x}_2|\boldsymbol{y}_{1:2}) \cdots p(\boldsymbol{x}_N|\boldsymbol{x}_{N-1}, \boldsymbol{y}_{1:N-1}) p(\boldsymbol{x}_{N-1}|\boldsymbol{y}_{1:N-1}) p(\boldsymbol{x}_N|\boldsymbol{y}_{1:N}) \tag{9.68}$$

$$= p(\boldsymbol{x}_2|\boldsymbol{x}_1) p(\boldsymbol{x}_1|\boldsymbol{y}_1) p(\boldsymbol{x}_3|\boldsymbol{x}_2) p(\boldsymbol{x}_2|\boldsymbol{y}_{1:2}) \cdots p(\boldsymbol{x}_N|\boldsymbol{x}_{N-1}) p(\boldsymbol{x}_{N-1}|\boldsymbol{y}_{1:N-1}) p(\boldsymbol{x}_N|\boldsymbol{y}_{1:N}) \tag{9.69}$$

$$= \text{Normal}\left(\boldsymbol{x}_2; B_2\boldsymbol{x}_1 + b_2, V_n\right) \text{Normal}\left(\boldsymbol{x}_1; r_{1|1}, R_{1|1}\right) \text{Normal}\left(\boldsymbol{x}_3; B_3\boldsymbol{x}_2 + b_3, V_n\right) \text{Normal}\left(\boldsymbol{x}_2; r_{2|2}, R_{2|2}\right) \cdots \text{Normal} \tag{9.70}$$

which suggest that $\boldsymbol{x}_N$ can be sampled from the terminal filter

$$\boldsymbol{x}_N \sim \text{Normal}\left(r_{N|N}, R_{N|N}\right) \tag{9.71}$$

and subsequently sample earlier states sequentially from

$$\boldsymbol{x}_n \sim \text{Normal}\left(\boldsymbol{x}_{n+1}; B_{n+1}\boldsymbol{x}_n + b_{n+1}, V_{n+1}\right) \text{Normal}\left(\boldsymbol{x}_n; r_{n|n}, R_{n|n}\right) \tag{9.72}$$

$$= \text{Normal}\left(\boldsymbol{x}_n; ???, ???\right), \qquad n < N \tag{9.73}$$

Example 9.2

Revisit the example with Brownian motion for the simplest case: linear forces and Gaussian noise.

9.2.2 Extended Kalman theory for weakly non-linear Gaussian models

In the case where (i) the prior is Gaussian, (ii) noises are additive and Gaussian, and (iii) dynamics and measurements are non-linear, the state-space representation is

$$\boldsymbol{x}_1 \sim \text{Normal}\,(c, W) \tag{9.74}$$
$$\boldsymbol{x}_n|\boldsymbol{x}_{n-1} \sim \text{Normal}\,(g_n(\boldsymbol{x}_{n-1}), V_n)\,, \qquad n = 2, 3, \ldots \tag{9.75}$$
$$\boldsymbol{y}_n|\boldsymbol{x}_n \sim \text{Normal}\,(f_n(\boldsymbol{x}_n), U_n)\,, \qquad n = 1, 2, \ldots \tag{9.76}$$

Unlike the simplest case presented above, this representation does not allow for analytic derivations of the involved densities. Nevertheless, when the functions $g_n(\boldsymbol{x})$ and $f_n(\boldsymbol{x})$ are fairly linear, an approximate theory that parallels the linear one can be developed.

The main characteristic is that for the computation of the involved densities (i.e. filter, smoother, etc), one now relies on a local approximation around selected points. In particular, dynamics $g_n(\boldsymbol{x})$ and observations $f_n(\boldsymbol{x})$ can linearized through Taylor expansions

$$g_n(\boldsymbol{x}) \approx g_n(\boldsymbol{x}^*) + \nabla_{\boldsymbol{x}} g_n(\boldsymbol{x}^*)\,(\boldsymbol{x} - \boldsymbol{x}^*) \tag{9.77}$$
$$f_n(\boldsymbol{x}) \approx f_n(\boldsymbol{x}^*) + \nabla_{\boldsymbol{x}} f_n(\boldsymbol{x}^*)\,(\boldsymbol{x} - \boldsymbol{x}^*) \tag{9.78}$$

Nevertheless, the precise location of the base-point $\boldsymbol{x}^*$ depends on the density to be computed. For example, for the computation of the filters, the standard choice is to linearize the dynamics around the mean of the previous filter and to linearize the observations around the mean of the predictions.

Here describe in detail only filter, forcaster, and smoother. These are the only standard approximations in the extended Kalman theory.

Example 9.3

Revisit the example with Brownian motion for: non-linear forces and Gaussian noise with and without PSF.

9.3 Beyond simple state-space models

In all state-space representations we have seen so far, we have made the silent assumption that every parameter that influences either the dynamics or the measurements are characterized well and every one of them attains values known in advance. While for some applications this may be the case, in most applications we have to estimate appropriate values for the parameters while we estimate the unknown states and so we need to modify the preceding theory.

In the presence of unknown parameters $\boldsymbol{\theta}$, an appropriate model to work with is the following

$$\boldsymbol{\theta} \sim I \tag{9.79}$$
$$\boldsymbol{x}_1 \sim H \tag{9.80}$$
$$\boldsymbol{x}_n|\boldsymbol{x}_{n-1}, \boldsymbol{\theta} \sim G^{\boldsymbol{\theta}}_{\boldsymbol{x}_{n-1}}, \qquad n = 2, 3, \ldots \tag{9.81}$$
$$\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta} \sim F^{\boldsymbol{\theta}}_{\boldsymbol{x}_n}, \qquad n = 1, 2, \ldots \tag{9.82}$$

where dynamics $G^{\boldsymbol{\theta}}_{\boldsymbol{x}_{n-1}}$ and measurement $F^{\boldsymbol{\theta}}_{\boldsymbol{x}_n}$ densities now depend on the parameters $\boldsymbol{\theta}$ and I is an appropriate prior on $\boldsymbol{\theta}$.

Algorithm 9.3: Gibbs sampling for state-space models with unknown parameters

To draw samples from the posterior $p(\boldsymbol{\theta}, \boldsymbol{x}_{1:N}|\boldsymbol{y}_{1:N})$ of the model in eqs. (9.79) to (9.82) initialize at some feasible values $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{x}_{1:N}^{(0)}$
for $i = 1, 2, \ldots$ repeat
(i) sample parameters $\boldsymbol{\theta}^{(i)}$ from $p(\boldsymbol{\theta}|\boldsymbol{x}_{1:N}^{(i-1)}, \boldsymbol{y}_{1:N})$
(ii) sample states $\boldsymbol{x}_{1:N}^{(i)}$ from $p(\boldsymbol{x}_{1:N}|\boldsymbol{\theta}^{(i)}, \boldsymbol{y}_{1:N})$

For those cases, there are at least two strategies to proceed with the analysis. Either to evaluate the posterior $p(\boldsymbol{\theta}, \boldsymbol{x}_{1:N}|\boldsymbol{y}_{1:N})$ using an MCMC approach, for example based on Gibbs sampling according to algorithm 9.3, or model the parameters $\boldsymbol{\theta}$ as dynamical variables $\boldsymbol{\theta}_n$ and apply the theory we have developed earlier for an augmented state $\boldsymbol{x}'_n = (\boldsymbol{x}_n, \boldsymbol{\theta}_n)$.

In particular, for the latter approach one needs to introduce *artificial dynamics* for the parameters that lead to *reasonable approximations*. For example, an approximate model that correspond to eqs. (9.79) to (9.82) can be

$$\boldsymbol{\theta}_1 \sim I \tag{9.83}$$

$$\boldsymbol{x}_1 \sim H \tag{9.84}$$

$$\boldsymbol{x}_n|\boldsymbol{x}_{n-1}, \boldsymbol{\theta}_{n-1} \sim G_{\boldsymbol{x}_{n-1}}^{\boldsymbol{\theta}_{n-1}}, \qquad n = 2, 3, \ldots \tag{9.85}$$

$$\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1} \sim \text{Normal}\left(\boldsymbol{\theta}_{n-1}, \epsilon^2\right) \tag{9.86}$$

$$\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta}_n \sim F_{\boldsymbol{x}_n}^{\boldsymbol{\theta}_n}, \qquad n = 1, 2, \ldots \tag{9.87}$$

For $\epsilon \ll 1$ the posterior $p(\theta_{1:N}, \boldsymbol{x}_{1:N}|\boldsymbol{y}_{1:N})$ corresponding to this model approaches the posterior $p(\theta, \boldsymbol{x}_{1:N}|\boldsymbol{y}_{1:N})$ of the initial model.

Example 9.4

Consider the case of Brownian motion in 1D that is observed under noisy measurements. As usual, consider x_n, y_n to be the true and measured position, respectively, at time t_n. Further, suppose the diffusion coefficient D and noise variance ω are unknown.

Chapter 10

Continuous time processes

Part III

Appendix

Appendix A

Notation and other conventions

A.1 Time and other physical quantities

Throughout this book, we adopt, to the degree possible, terminology that make sense in the broader context of Natural Sciences. For example, in the dynamical systems or machine learning literature it is customary to present "time" in a unit-less integer-valued fashion. Unfortunately, such approach might be quite confusing in our physical context and we discourage it.

We also note that we pay special attention in maintaining physically correct units. So, in the examples and exercises, temporal variables are measured in units of time, spatial variables are measured in units of space and so on.

Starting in chapter 2, time becomes the most important notion and for this reason we reserve the most important indices for temporal quantities. We generally denote time with subscripts, although some exceptions persist. As real-life experiments provide temporally arranged measurements, we use the same convention to index individual data-points even when our main interest is focus on time independent problems.

A.2 Random variables and other mathematical notions

Starting in **??**, we generally distinguish between a random variable, most often by using capital letters, over the values that a random variable takes, most often by using literal letters. Additionally, we mostly distinguish between probability distributions and probability densities, although in this case we are not very consistent in following this rule across all chapters.

We generally denote with bold faced letters vector or matrix quantities. As we explain in detail below, we are careful to distinguish between lists, vectors, and arrays and for all these we adopt different notation.

A.3 Collections

Even from **??**, we encounter models consisting of numerous variables and it is quite practical to arrange them into groups. Here, we gather our conventions concerning such grouping.

- When the ordering of the variables is *unimportant*, we will denote groups with braces, for example $\{\alpha, \beta, \gamma\}$, and refer to them as *lists*. In these cases, it is valid to write down expressions like

$$\{\alpha, \beta, \gamma\} = \{\alpha, \gamma, \beta\}, \qquad \{1, 2, 3\} = \{1, 3, 2\}.$$

  In the particular case of lists of *indexed* variables, for example $w_1, w_2, \cdots, w_N$, instead of $\{w_1, w_2, \cdots, w_N\}$, often we will be using a compact notation with subscripts $w_{1:N}$ and it will be valid to write down

$$w_{1:N} = \{w_1, w_2, \cdots, w_N\}.$$

- When the ordering of the variables is *important*, we will denote groups with parenthesis, for example (x, y, z), and refer to them as *vectors*. In these cases, it is valid to write down expressions like

$$(x, y, z) \neq (x, z, y), \qquad (1, 2, 3) \neq (1, 3, 2).$$

- We will also encounter cases where the ordering of indexed variables *depends on the indices*, for example variables like $\pi_{\rho_1}, \pi_{\rho_2}, \cdots, \pi_{\rho_M}$. We will denote such groups with brackets with the ordering explicitly shown, for example ${}_{\rho_1,\rho_2,\cdots,\rho_M}[\pi_{\rho_1}, \pi_{\rho_2}, \cdots, \pi_{\rho_M}]$, and refer to them as *arrays*. In these cases, it is valid to write down expressions like

$$ {}_{\sigma_1,\sigma_2,\sigma_3}[1,2,3] \neq {}_{\sigma_1,\sigma_2,\sigma_3}[1,3,2], \qquad\qquad {}_{\sigma_1,\sigma_2,\sigma_3}[1,2,3] = {}_{\sigma_1,\sigma_3,\sigma_2}[1,3,2]. $$

  Of course, when the ordering adopted is obvious, generally, we will avoid showing it and simply write $[\pi_{\sigma_1}, \pi_{\sigma_2}, \cdots, \pi_{\sigma_M}]$ instead of the more elaborate ${}_{\rho_1,\rho_2,\cdots,\rho_M}[\pi_{\rho_1}, \pi_{\rho_2}, \cdots, \pi_{\rho_M}]$.

Appendix B

Numerical random variables

TODO: change random variables from X to R. Add Dirichlet

Below we list most common scalar random variables. In this presentation we use unified notation where $\mathbb{P}(\boldsymbol{\theta})$ denotes a probability distribution that depends on $\boldsymbol{\theta}$ which, in general, might include one or more parameters. For a random variable $X \sim \mathbb{P}(\boldsymbol{\theta})$ attaining values x, we use $p(x;\boldsymbol{\theta})$ to denote its probability density, which also depends on $\boldsymbol{\theta}$.

The support of a random variable $X \sim \mathbb{P}(\boldsymbol{\theta})$ gathers all values x that X may attain. Of course, the associated density $p(x;\boldsymbol{\theta})$ is non-zero when x is in the support of $\mathbb{P}(\boldsymbol{\theta})$ and zero otherwise.

As with all random variables attaining a probability density, probabilities are obtained by summing $dx\, p(x;\boldsymbol{\theta})$, which results in appropriate integrals of $p(x;\boldsymbol{\theta})$. This indicates that the density $p(x;\boldsymbol{\theta})$ has *units* and these equal to the *inverse units* of dx.

To avoid confusion with the various parameterizations of the distributions, for each random variable below we also list its first and second moments. For a random variable $X \sim \mathbb{P}(\boldsymbol{\theta})$, we use $\mathbb{E}(X)$ to denote its mean and $\mathbb{V}(X)$ to denote its variance. These are computed by the integrals

$$\mathbb{E}(X) = \int dx\, p(x;\boldsymbol{\theta}) x, \qquad \mathbb{V}(X) = \int dx\, p(x;\boldsymbol{\theta})\, (x - \mathbb{E}(X))^2 .$$

Often, certain parameters in $\boldsymbol{\theta}$ characterize the spread of the values x over the support of $X \sim \mathbb{P}(\boldsymbol{\theta})$. We refer to these parameters as *scales* when they have the same units as the mean of X and as *rates* when they have the inverse units.

Note B.1: Uniform random variable

Definition A uniform random variable $X \sim \text{Uniform}_{[x_{min}, x_{max}]}$, takes real scalar values x, between x_{min} and x_{max}, and has the probability density and moments

$$\text{Uniform}_{[x_{min}, x_{max}]}(x) = \frac{1}{x_{max} - x_{min}}, \qquad \mathbb{E}(X) = \frac{x_{max} + x_{min}}{2}, \qquad \mathbb{V}(X) = \frac{(x_{max} + x_{min})^2}{12}.$$

Parameterization In the common parametrization, the *end-points* x_{min} and x_{max} are real numbers.

Sampling scheme To simulate $X \sim \text{Uniform}_{[x_{min}, x_{max}]}$, first simulate $U \sim \text{Uniform}_{[0,1]}$, and set $x = x_{min} + (x_{max} - x_{min})u$. A command to simulate U is offered in most software packages.

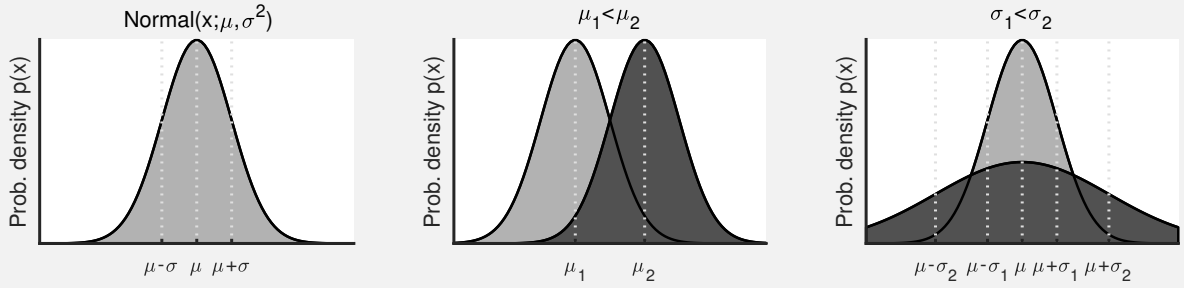


Figure B.1: The normal distribution.

Note B.2: Normal random variable

Definition A normal random variable $X \sim \text{Normal}(\mu, v)$, also termed Gaussian, takes real scalar values x and has the probability density and moments

$$\text{Normal}(x; \mu, v) = \frac{1}{\sqrt{2\pi v}} \exp\left(-\frac{(x-\mu)^2}{2v}\right), \qquad \mathbb{E}(X) = \mu, \qquad \mathbb{V}(X) = v.$$

Parameterization In the common parametrization, the *mean* μ and *variance* v are real numbers and v is positive.

Alternative parametrization Alternative parameterizations include mean μ and standard deviation $\sigma = \sqrt{v}$, or mean μ and precision $\tau = 1/v$. According to these

$$\text{Normal}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \qquad \mathbb{E}(X) = \mu, \qquad \mathbb{V}(X) = \sigma^2,$$

$$\text{Normal}\left(x; \mu, \frac{1}{\tau}\right) = \sqrt{\frac{\tau}{2\pi}} \exp\left(-\frac{\tau(x-\mu)^2}{2}\right), \qquad \mathbb{E}(X) = \mu, \qquad \mathbb{V}(X) = \frac{1}{\tau}.$$

Related distributions The $\text{Normal}(\mu, v)$ distribution is the special of case of the $\text{MNormal}_1(x; \mu, v)$ distribution.

Sampling scheme To simulate $X \sim \text{Normal}(\mu, v)$, first simulate $U \sim \text{Normal}(0, 1)$, and set $x = \mu + \sqrt{v}u$. A command to simulate U is offered in most software packages.

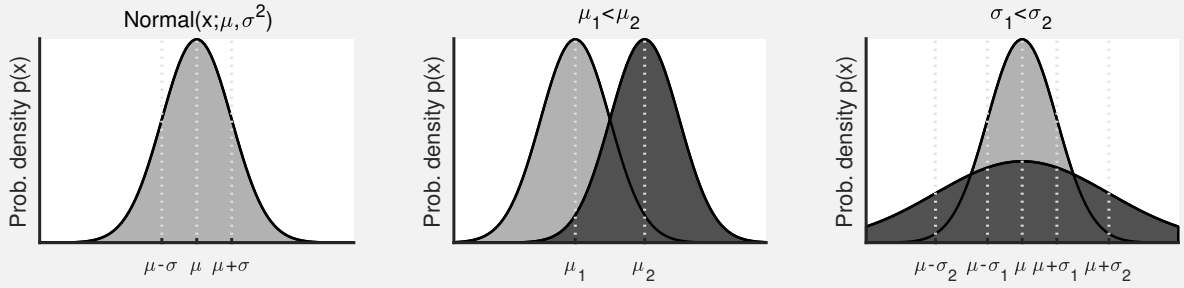


Figure B.2: The normal distribution.

Note B.3: Student-T random variable

Definition A Student-T random variable $X \sim \text{StudentT}_\nu(\mu, \sigma)$, takes real scalar values x and has the probability

density and moments

$$\text{StudentT}(x;\mu,\sigma) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}\frac{1}{\sqrt{\pi\nu}}\frac{1}{\sigma}\left(1+\frac{1}{\nu}\left(\frac{x-\mu}{\sigma}\right)^2\right)^{-\frac{\nu+1}{2}}, \qquad \mathbb{E}(X)=\mu, \qquad \mathbb{V}(X)=\frac{\nu}{\nu-2}\sigma^2,$$

where $\Gamma(\nu)$ is the Gamma function.

Parameterization In the common parametrization, the *mean* μ, *scale* σ, and degrees of freedom ν are real numbers and ν and σ are positive.

Related distributions The $\text{Cauchy}(\mu,\sigma)$ distribution is a special of case of the $\text{StudentT}_1(\mu,\sigma)$ distribution.

Sampling scheme To simulate $X \sim \text{StudentT}_\nu(\mu,\sigma)$, first simulate $U_1 \sim \text{Normal}(0,1)$ and $U_2 \sim \text{Gamma}(\nu/2,1)$, and set $x = \mu + \sigma\sqrt{\frac{\nu}{2u_2}}u_1$. A command to simulate U_1 and U_2 is offered in most software packages.

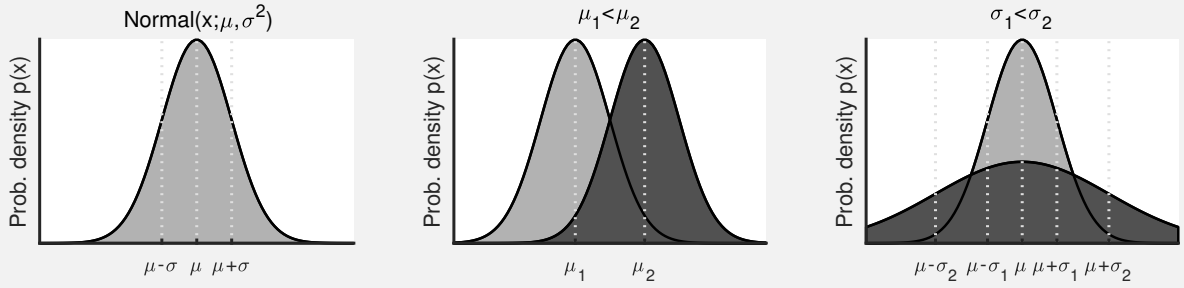


Figure B.3: The normal distribution.

Note B.4: Cauchy random variable

Definition A Cauchy random variable $X \sim \text{Cauchy}(\mu,\sigma)$, takes real scalar values x and has the probability density and moments

$$\text{Cauchy}(x;\mu,\sigma) = \frac{1}{\pi}\frac{\sigma}{\sigma^2+(x-\mu)^2}, \qquad \mathbb{E}(X)=\infty, \qquad \mathbb{V}(X)=\infty.$$

Parameterization In the common parametrization, the *mean* μ and *scale* σ are real numbers and σ is positive.

Related distributions The $\text{Cauchy}(\mu,\sigma)$ distribution is a special of case of the $\text{StudentT}_1(\mu,\sigma)$ distribution.

Sampling scheme To simulate $X \sim \text{Cauchy}(\mu,\sigma)$, first simulate $U \sim \text{Uniform}_{[0,1]}$, and set $x = \mu+\sigma\tan\left(\pi(u-\frac{1}{2})\right)$. A command to simulate U is offered in most software packages.

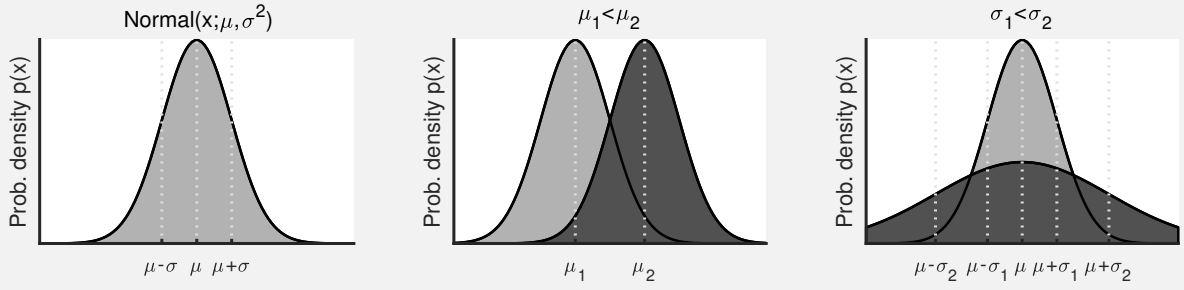


Figure B.4: The normal distribution.

Note B.5: Exponential random variable

Definition An exponential random variable $X \sim \text{Exponential}(\lambda)$ takes positive real scalar values x and has the probability density and moments

$$\text{Exponential}(x; \lambda) = \lambda e^{-\lambda x}, \qquad \mathbb{E}(X) = \frac{1}{\lambda}, \qquad \mathbb{V}(X) = \frac{1}{\lambda^2}.$$

Parameterization In the common parametrization, the *rate* λ is a positive real number.

Related distributions Alternative parameterizations include mean or scale $\mu = 1/\lambda$. According to this

$$\text{Exponential}\left(x; \frac{1}{\mu}\right) = \frac{1}{\mu} e^{-\frac{x}{\mu}}, \qquad \mathbb{E}(X) = \mu, \qquad \mathbb{V}(X) = \mu^2.$$

Related distributions The $\text{Exponential}(\lambda)$ distribution is a special of case of the $\text{Gamma}(1, 1/\lambda)$ distribution.

Sampling scheme To simulate $X \sim \text{Exponential}(\lambda)$, first simulate $U \sim \text{Uniform}_{[0,1]}$, and set $x = -\frac{1}{\lambda} \log u$. A command to simulate U is offered in most software packages.

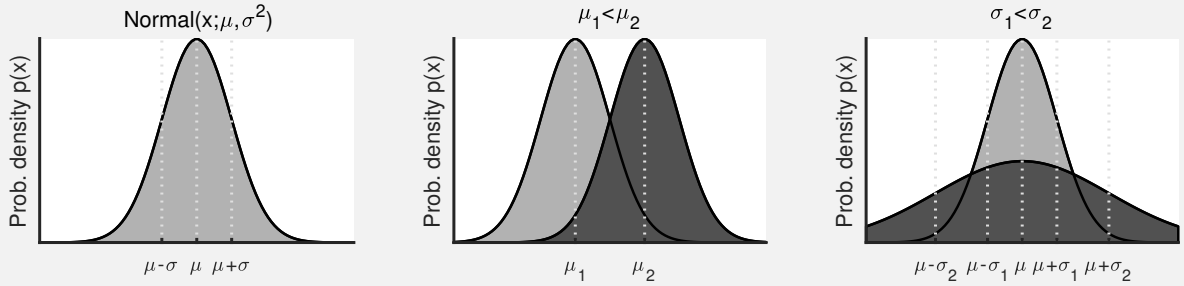


Figure B.5: The normal distribution.

Note B.6: Gamma random variable

Definition A gamma random variable $X \sim \text{Gamma}(\alpha, \beta)$ takes positive real scalar values x and has the probability density and moments

$$\text{Gamma}(x; \alpha, \beta) = \frac{1}{\beta \Gamma(\alpha)} \left(\frac{x}{\beta}\right)^{\alpha - 1} e^{-\frac{x}{\beta}}, \qquad \mathbb{E}(X) = \alpha\beta, \qquad \mathbb{V}(X) = \alpha\beta^2.$$

where $\Gamma(\alpha)$ is the Gamma function.

Parameterization In the common parametrization, the *shape* α and *scale* are positive real numbers.

Related distributions The $\text{Exponential}(\lambda)$ distribution is a special of case of the $\text{Gamma}(1, 1/\lambda)$ distribution.

Sampling scheme To simulate $X \sim \text{Gamma}(\alpha, \beta)$, first simulate $U \sim \text{Gamma}(\alpha, 1)$, and set $x = \beta u$. A command to simulate U is offered in most software packages.

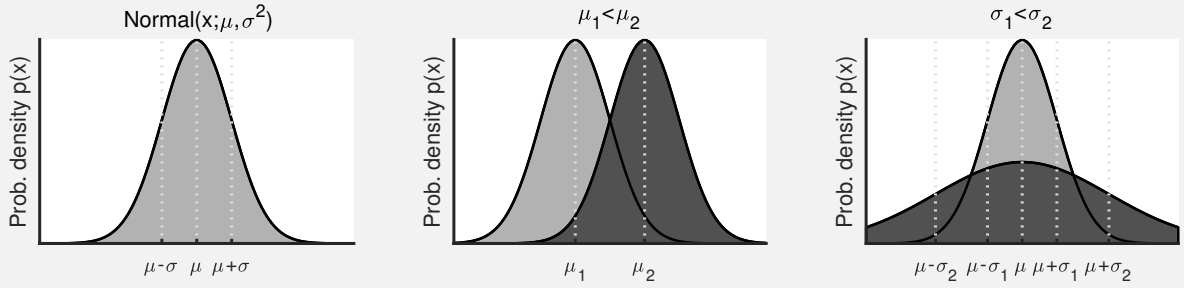


Figure B.6: The normal distribution.

Note B.7: Beta random variable

Definition A beta random variable $X \sim \text{Beta}(\alpha, \beta)$ takes positive real scalar values x between 0 and 1 and has the probability density and moments

$$\text{Beta}(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}, \qquad \mathbb{E}(X) = \frac{\alpha}{\alpha+\beta}, \qquad \mathbb{V}(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}.$$

where $B(\alpha, \beta)$ is the Beta function.

Parameterization In the common parametrization, the *shapes* α and β are positive real numbers.

Related distributions The $\text{Uniform}_{[0,1]}$ distribution is a special of case of the $\text{Beta}(1, 1)$ distribution.

Sampling scheme To simulate $X \sim \text{Beta}(\alpha, \beta)$, first simulate $U_1 \sim \text{Gamma}(\alpha, 1)$, $U_2 \sim \text{Gamma}(\beta, 1)$, and set $x = u_1/(u_1 + u_2)$. A command to simulate U_1, U_2 is offered in most software packages.

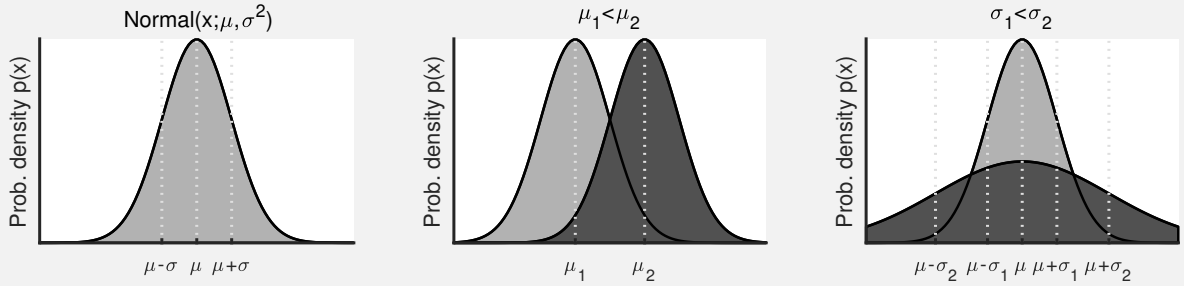


Figure B.7: The normal distribution.

Note B.8: Beta-prime random variable

Definition A beta-prime random variable $X \sim \text{BetaPrime}(\alpha, \beta)$ takes positive real scalar values x and has the probability density and moments

$$\text{BetaPrime}(x; \alpha, \beta) = \frac{x^{\alpha-1}(1+x)^{-\alpha-\beta}}{B(\alpha, \beta)}, \qquad \mathbb{E}(X) = \frac{\alpha}{\beta-1}, \qquad \mathbb{V}(X) = \frac{\alpha(\alpha+\beta-1)}{(\beta-2)(\beta-1)^2}.$$

where $B(\alpha, \beta)$ is the Beta function.

Parameterization In the common parametrization, the *shapes* α and β are positive real numbers.

Sampling scheme To simulate $X \sim \text{BetaPrime}(\alpha, \beta)$, first simulate $U_1 \sim \text{Gamma}(\alpha, 1)$, $U_2 \sim \text{Gamma}(\beta, 1)$, and set $x = u_1/u_2$. A command to simulate U_1, U_2 is offered in most software packages.

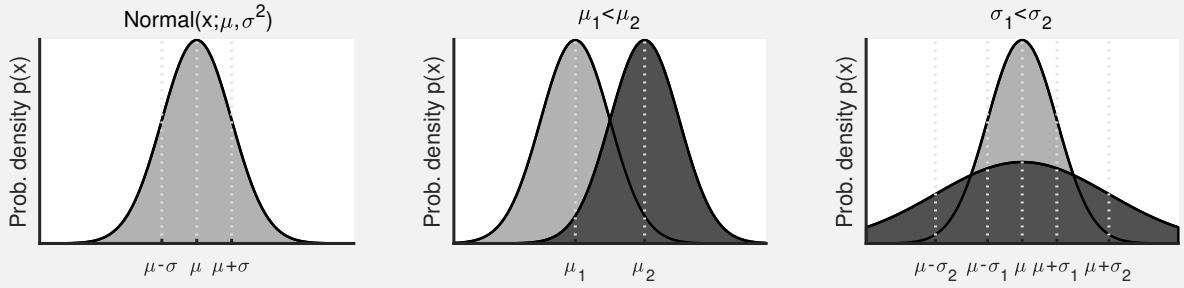


Figure B.8: The normal distribution.

Note B.9: Poisson random variable

Definition A Poisson random variable $X \sim \text{Poisson}(\mu)$, takes non-negative integer scalar values x and has the probability density and moments

$$\text{Poisson}(x;\mu) = \sum_{k=0}^{\infty} \exp(-\mu)\frac{\mu^k}{k!}\,\delta_k(x), \qquad \mathbb{E}(X) = \mu, \qquad \mathbb{V}(X) = \mu.$$

Parameterization In the common parametrization, the *mean* μ is a real number.

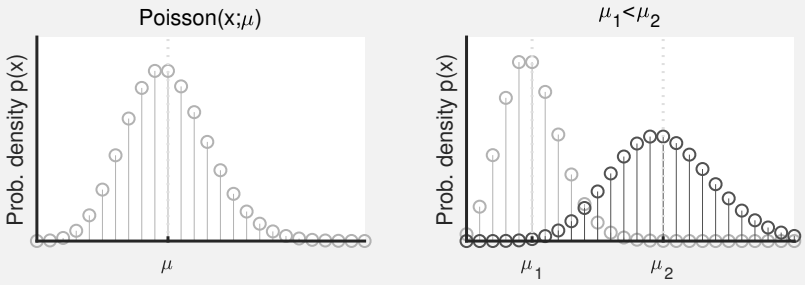


Figure B.9: The Poisson distribution.

Note B.10: Geometric random variable

Definition A geometric random variable $X \sim \text{Geometric}(\pi)$, takes non-negative integer scalar values x and has the probability density and moments

$$\text{Geometric}(x;\pi) = \sum_{k=0}^{\infty}(1-\pi)^k\pi\,\delta_k(x), \qquad \mathbb{E}(X) = \frac{1-\pi}{\pi}, \qquad \mathbb{V}(X) = \frac{1-\pi}{\pi^2}.$$

Parameterization In the common parametrization, the *success probability* π is a real number between 0 and 1. Show the parametrization that starts at 0. Also show the relationship with NegBinomial

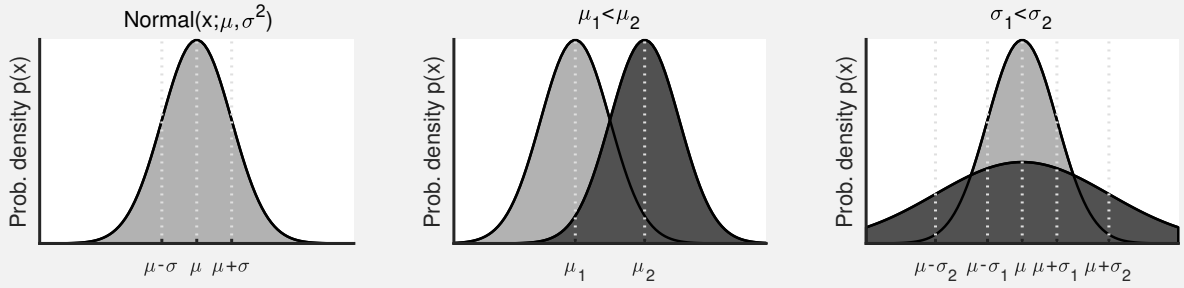


Figure B.10: The normal distribution.

Note B.11: Binomial random variable

Definition A binomial random variable $X \sim \text{Binomial}(n, \pi)$, takes non-negative integer scalar values x and has the probability density and moments

$$\text{Binomial}(x; n, \pi) = \sum_{k=0}^{\infty} \frac{n!}{k!(n-k)!} \pi^k (1-\pi)^{n-k} \, \delta_k(x), \qquad \mathbb{E}(X) = n\pi, \qquad \mathbb{V}(X) = n\pi(1-\pi).$$

Parameterization In the common parametrization, the *number of trials* n is an integer and the *success probability* π is a real number between 0 and 1.

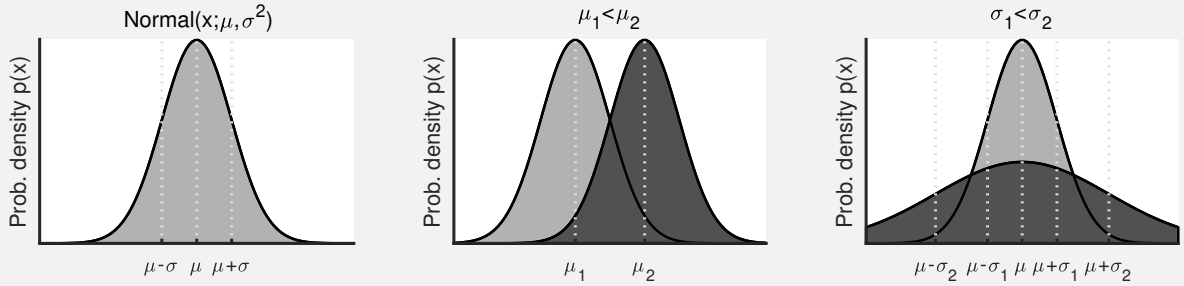


Figure B.11: The normal distribution.

Note B.12: Negative-Binomial random variable

Definition A negative-binomial random variable $X \sim \text{NegBinomial}(n, \pi)$, takes non-negative integer scalar values x and has the probability density and moments

$$\text{NegBinomial}(x; n, \pi) = \sum_{k=0}^{\infty} \pi^x (1-\pi) \, \delta_k(x), \qquad \mathbb{E}(X) = ????, \qquad \mathbb{V}(X) = ????.$$

Parameterization In the common parametrization, the *????* n is an integer and the *????* π is a real number between 0 and 1. show relation with Geometric

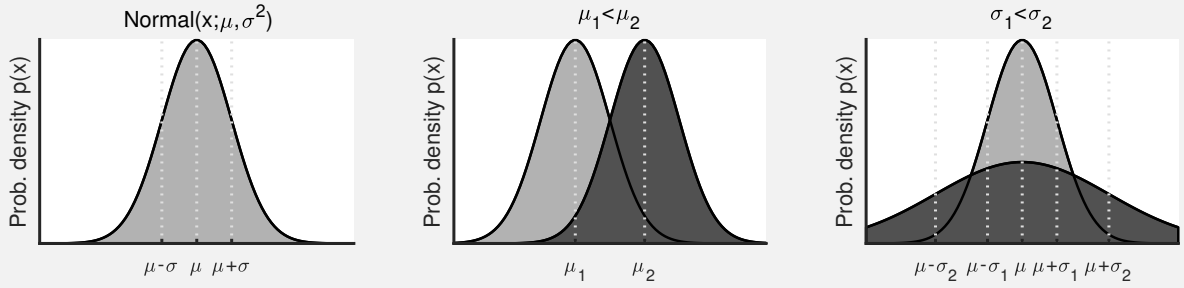


Figure B.12: The normal distribution.

Note B.13: Beta-Negative-Binomial random variable

Definition A beta-negative-binomial random variable $X \sim$ BetaNegBinomial(n, α, β), takes non-negative integer scalar values x and has the probability density and moments

$$\text{BetaNegBinomial}(x; \alpha, \beta) = \sum_{k=0}^{\infty} \beta \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)} \frac{\Gamma(\alpha+x)}{\Gamma(1+\alpha+\beta+x)} \delta_k(x), \qquad \mathbb{E}(X) = \frac{\alpha}{\beta-1}, \qquad \mathbb{V}(X) = ????.$$

Parameterization In the common parametrization, the *????* n is an integer and the *????* π is a real number between 0 and 1. show relation with Geometric

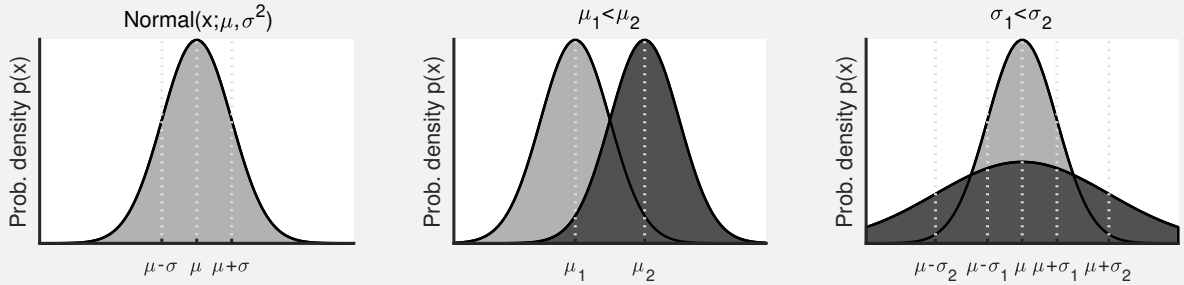


Figure B.13: The normal distribution.

Note B.14: Bernoulli random variable

Definition A Bernoulli random variable $X \sim$ Bernoulli(π), takes values 1 or 0 and has the probability density and moments

$$\text{Bernoulli}(x; \pi) = \pi\delta_1(x) + (1-\pi)\delta_0(x), \qquad \mathbb{E}(X) = \pi, \qquad \mathbb{V}(X) = \pi(1-\pi).$$

Parameterization In the common parametrization, the *success probability* π is a real number between 0 and 1.

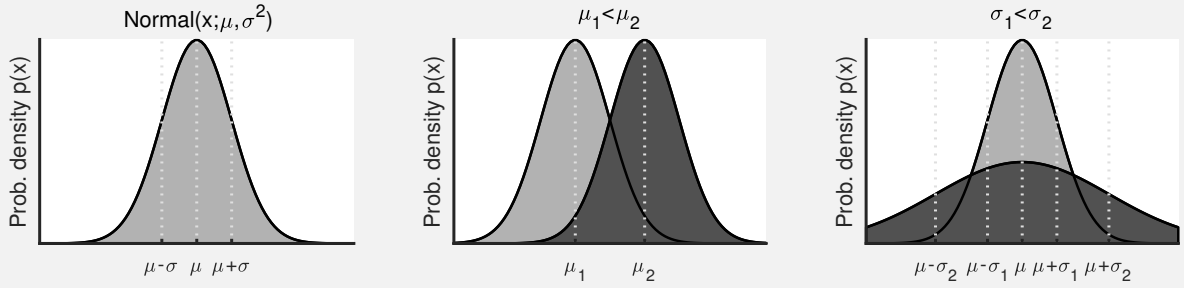


Figure B.14: The normal distribution.

Appendix C

The Dirac δ

The Dirac δ firstly appears in section 1.2 when we describe probability densities over discrete random variables. We denote the Dirac δ by $\delta_\rho(r)$.

Note C.1: Notation

Our notation differs slightly from that used in Physics where $\delta_\rho(r)$ would normally be written as $\delta(r-\rho)$. In physics, $\delta(\rho - r)$ is used to represent how mass, charge, or other quantities are spread over time or space. In this setting, the variables r and ρ typically denote points in space or time for which subtraction $r-\rho$ is meaningful.

Here, however, $\delta_\rho(r)$ represents how probability is spread over the values of the random variables of interest. In the probabilistic context, r and ρ may stand for any quantity, even for such quantities where subtraction may be meaningless, such as the constitutive states in a system's state-space, chapter 2. For this reason, to denote a Dirac δ centered at an arbitrary point ρ, we prefer the more general notation $\delta_\rho(r)$ instead of $\delta(r-\rho)$.

C.1 Definition

In one dimension, a common description of the Dirac δ is through the limit of increasingly thiner Gaussians or Normal densities

$$\delta_\rho(r) = \lim_{\sigma\to 0} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(r-\rho)^2}{2\sigma^2}} = \lim_{\sigma\to 0} \text{Normal}\left(r;\rho,\sigma^2\right), \qquad r\neq\rho. \tag{C.1}$$

Many descriptions equivalent to eq. (C.1) also exist. For instance, the Gaussians are replaced by Lorentzians or Cauchy densities

$$\delta_\rho(r) = \lim_{\sigma\to 0} \frac{1}{\pi}\frac{\sigma}{\sigma^2+(r-\rho)^2} = \lim_{\sigma\to 0} \text{Cauchy}\left(r;\rho,\sigma\right), \qquad r\neq\rho. \tag{C.2}$$

Descriptions of this type are quite intuitive, for example see fig. C.1, and are readily extended to more than one dimensions by replacing Normal or Cauchy with the appropriate multivariate extensions.

We note that in eqs. (C.1) and (C.2) the integrals yield $\int_{-\infty}^{+\infty} dr\, \text{Normal}\left(r;\rho,\sigma^2\right) = 1$ or $\int_{-\infty}^{+\infty} dr\, \text{Cauchy}\left(r;\rho,\sigma\right) = 1$ and the values of the integral remain constant irrespective of σ whose value is infinitesimally small. This property also carries over to the $\sigma\to 0$ limit and applies to $\delta_\rho(r)$ as well. For instance,

$$\int_{-\infty}^{+\infty} dr\, \delta_\rho(r) = \int_{-\infty}^{+\infty} dr\, \lim_{\sigma\to 0} \text{Normal}\left(r;\rho,\sigma^2\right) = \lim_{\sigma\to 0}\int_{-\infty}^{+\infty} dr\, \text{Normal}\left(r;\rho,\sigma^2\right) = \lim_{\sigma\to 0} 1 = 1.$$

This last equality, combined with an explicit evaluation of the limits $\sigma\to 0$, namely

$$\delta_\rho(r) = 0, \qquad r\neq\rho \tag{C.3}$$

$$\int_{\substack{\text{all values}\\ \text{of } r}} dr\, \delta_\rho(r) = 1 \tag{C.4}$$

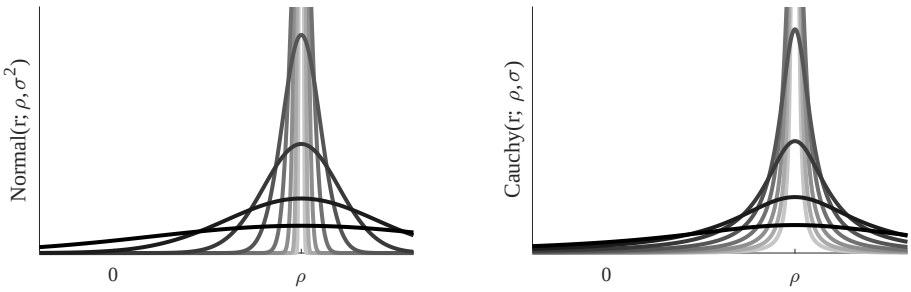


Figure C.1: The Dirac $\delta_\rho(r)$ as a sequence of successively thinner probability densities.

are perhaps the most distinctive characteristics of the Dirac δ. Since eqs. (C.3) and (C.4) are independent of Normal or Cauchy appearing in eqs. (C.1) and (C.2), they offer a better starting point for the definition of Dirac δ.

As special cases, we invoke eqs. (C.1) and (C.2) only in situations where subtractions like $r-\rho$ actually make sense. These include cases where r, ρ are scalars or vectors. In more general settings, however, the definition of $\delta_\rho(r)$ can be described only abstractly. In particular, in an abstract setting, it is sufficient to start directly from the conditions eqs. (C.3) and (C.4) without any reference to an underlying limiting scheme.

Note C.2: What is $\delta_\rho(\rho)$?

Motivated by eqs. (C.1) and (C.2), which provide valid values for the limits even at $r=\rho$, *heuristic* definitions of $\delta_\rho(r)$ often assign a value of ∞ for $\delta_\rho(\rho)$. For instance, a typical heuristic is as follows

$$\delta_\rho(r) = \begin{cases} 0, & r \neq \rho. \\ \infty, & r = \rho. \end{cases}$$

However, specifying a value for $\delta_\rho(r)$ at $r=\rho$ is *unnecessary*. In fact, assigning a value for $\delta_\rho(\rho)$ and in particular ∞ is misleading and may lead to inaccuracies. For example, since $2\times 0 = 0$ and $2\times\infty = \infty$, a reasoning stream may proceed as follows

$$1 = \int_{-\infty}^{+\infty} dr\, \delta_\rho(r) = \int_{-\infty}^{+\infty} dr\, 2\delta_\rho(r) = 2\int_{-\infty}^{+\infty} dr\, \delta_\rho(r) = 2.$$

While eqs. (C.3) and (C.4) are *necessary;* to avoid paradoxes like this, it is much safer to leave $\delta_\rho(\rho)$ defined abstractly. This way, $\delta_\rho(\rho)$ remains free to attain any value that is needed to make eqs. (C.3) and (C.4) consistent.

Of course, values of $\delta_\rho(\rho)$ that make eqs. (C.3) and (C.4) consistent with each other, *cannot* be numeric, finite or infinite. For this reason, the Dirac δ is designated as a *generalized* function, as it attains *non-numeric* values in a very significant manner.

C.2 Properties

As we see, $\delta_\rho(r)$ evaluates to 0 when $r \neq \rho$. This means that, when integrating over a domain $\mathcal{R}$ that does *not* include ρ, it evaluates to zero. However, when integrating over a domain $\mathcal{R}$, which *does* include ρ, it evaluates to unity. That is,

$$\int_{\mathcal{R}} dr\, \delta_\rho(r) = \begin{cases} 0, & \rho \notin \mathcal{R} \\ 1, & \rho \in \mathcal{R} \end{cases}$$

The same property also carries over when $\delta_\rho(r)$ is multiplied by a *continuous* function

$$\int_{\mathcal{R}} dr\, f(r)\delta_\rho(r) = \begin{cases} 0, & \rho \notin \mathcal{R} \\ f(\rho), & \rho \in \mathcal{R} \end{cases} \tag{C.5}$$

Note C.3: Notation

When $\mathcal{R}$ includes every possible value of r, eq. (C.5) is commonly written as

$$f\delta_\rho = f(\rho), \qquad\qquad \langle f, \delta_\rho \rangle = f(\rho),$$

or, to emphasize the functional nature of such expressions, as $f(\cdot)\delta_\rho(\cdot) = f(\rho)$ and $\langle f(\cdot), \delta_\rho(\cdot) \rangle = f(\rho)$. In addition, if the subtraction $r - \rho$ is meaningful, eq. (C.5) takes a more familiar form

$$(f * \delta_\rho)(r) = f(\rho)$$

where $*$ denotes the convolution.

For those r and ρ that multiplication with a scalar is meaningful, integral rescaling immediately yields

$$\delta_{\lambda\rho}(\lambda r) = \frac{\delta_\rho(r)}{|\lambda|^d}$$

where λ is a non-zero scalar and d is the dimension of r and ρ.

Appendix D

Memoryless distributions

In the context of the discrete state-space systems evolving in continuous time of section 2.3 and, in particular, of the Markov jump processes of section 2.3.2, *memorylessness* means that the probability of sampling a particular holding period h given that this period exceeds *any* threshold d, is the same as sampling a holding period equal to $h - d$ in the first place. In other words, memorylessness entails that the distribution of holding periods forgets the elapsed time represented by d.

Informally, the memorylessness requirement reads

$$\text{Probability of } (H > h \text{ given } H > d) = \text{Probability of } (H > h - d)\,.$$

To express this requirement formally, we will use $p(h)$ to denote the probability density of H. Under $p(h)$, we see that the right hand side equals to $\int_{h-d}^{\infty} d\eta\, p(\eta)$. Further, by the definition of conditional probability, the left hand side is the same as the ratio

$$\frac{\text{Probability of } (H > h \text{ and } H > d)}{\text{Probability of } (H > d)} = \frac{\text{Probability of } (H > h)}{\text{Probability of } (H > d)}$$

which, under $p(h)$, is equal to $\frac{\int_h^{\infty} d\eta\, p(\eta)}{\int_d^{\infty} d\eta\, p(\eta)}$. Therefore, our requirement formally reads

$$\frac{\int_h^{\infty} d\eta\, p(\eta)}{\int_d^{\infty} d\eta\, p(\eta)} = \int_{h-d}^{\infty} d\eta\, p(\eta)$$

which, after rearrangement of the terms, turns to

$$\int_h^{\infty} d\eta\, p(\eta) = \int_d^{\infty} d\eta\, p(\eta) \int_{h-d}^{\infty} d\eta\, p(\eta).$$

Differentiating once with respect to h, we obtain

$$p(h) = p(h-d) \int_d^{\infty} d\eta\, p(\eta).$$

Similarly differentiating once with respect to d, we obtain

$$-p(d) \int_{\infty}^{h-d} d\eta\, p(\eta) = p(h-d) \int_{\infty}^{d} d\eta\, p(\eta).$$

Equating the right hand sides of the last two equations, we obtain

$$p(h) = -p(d) \int_{\infty}^{h-d} d\eta\, p(\eta).$$

Finally, setting $d = 0$ and differentiating once more with respect to h, we obtain

$$p'(h) = -p(0)p(h).$$

The general solution of this differential equation is

$$p(h) = Ce^{-p(0)h}.$$

Normalization $\int_0^\infty dh\ p(h) = 1$, gives $C = p(0)$. Therefore, a memoryless probability density attains the form

$$p(h) = p(0)e^{-p(0)h}.$$

Of course, this is the density of an *exponential* probability distribution with rate, which we commonly denote λ, equal to $p(0)$.

The converse is also true. Namely, an exponential random variable $H \sim \text{Exponential}(\lambda)$ is memoryless. Indeed, we can verify this directly

$$\frac{\int_h^\infty d\eta\, \text{Exponential}(\eta;\lambda)}{\int_d^\infty d\eta\, \text{Exponential}(\eta;\lambda)} = \frac{1-e^{-\lambda h}}{1-e^{-\lambda d}} = 1 - e^{-\lambda(h-d)} = \int_{h-d}^\infty d\eta\, \text{Exponential}(\eta;\lambda).$$

Note D.1: Two properties of memoryless random variables

For a collection of memoryless random variables

$$H_m \sim \text{Exponential}(\lambda_m), \qquad m = 1, \ldots, M$$

the minimum $H_* = \min_{m=1,\ldots,M} H_m$ and the identity of the minimum $M_* = \operatorname{argmin}_{m=1,\ldots,M} H_m$ are also random variables. These are distributed according to

$$H_* \sim \text{Exponential}\left(\sum_{m=1}^M \lambda_m\right)$$

$$M_* \sim \text{Categorical}_{1:M}\left(\frac{\lambda_1}{\sum_{m=1}^M \lambda_m}, \cdots, \frac{\lambda_M}{\sum_{m=1}^M \lambda_m}\right)$$

We leave a proof of these as an exercise.

Appendix E

Derivation of key relations

E.0.1 Relations of section 8.3.1

Derivation of eq. (*8.10*)

$$
\begin{aligned}
\mathcal{A}_n(s_n) &= p(\boldsymbol{w}_{1:n}, s_n|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= \sum_{s_{n-1}} p(\boldsymbol{w}_{1:n}, s_{n-1}, s_n|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= \sum_{s_{n-1}} p(w_n|\boldsymbol{w}_{1:n-1}, s_{n-1}, s_n, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) p(s_n|\boldsymbol{w}_{1:n-1}, s_{n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) p(\boldsymbol{w}_{1:n-1}, s_{n-1}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= \sum_{s_{n-1}} p(w_n|s_n, \boldsymbol{\phi}) p(s_n|s_{n-1}, \boldsymbol{\Pi}) p(\boldsymbol{w}_{1:n-1}, s_{n-1}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= p(w_n|s_n, \boldsymbol{\phi}) \sum_{s_{n-1}} p(s_n|s_{n-1}, \boldsymbol{\Pi}) p(\boldsymbol{w}_{1:n-1}, s_{n-1}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= G\left(w_n; \phi_{s_n}\right) \sum_{s_{n-1}} \pi_{s_{n-1} \to s_n} \mathcal{A}_{n-1}(s_{n-1})
\end{aligned}
$$

Derivation of eq. (*8.11*)

$$
\begin{aligned}
\mathcal{A}_1(s_1) &= p(w_1, s_1|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= p(w_1|s_1, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) p(s_1|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= p(w_1|s_1, \boldsymbol{\phi}) p(s_1|\boldsymbol{\rho}) \\
&= G\left(w_1; \phi_{s_1}\right) \rho_{s_1}.
\end{aligned}
$$

E.0.2 Relations of section 8.3.2

Derivation of eq. (*8.12*)

$$
\begin{aligned}
p(s_N|\boldsymbol{w}_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) &= \frac{p(\boldsymbol{w}_{1:N}, s_N|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})}{p(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} \\
&= \frac{\mathcal{A}_N(s_N)}{p(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} \\
&\propto \mathcal{A}_N(s_N)
\end{aligned}
$$

Derivation of eq. (*8.13*)

$$\begin{aligned}
p\left(s_n|\boldsymbol{w}_{1:N},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right) &= \frac{p\left(\boldsymbol{w}_{1:n},s_n,\boldsymbol{w}_{n+1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)} \\
&= \frac{p\left(\boldsymbol{w}_{1:n},s_n|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)p\left(\boldsymbol{w}_{n+1:N}|\boldsymbol{w}_{1:n},s_n,\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)} \\
&= \frac{p\left(\boldsymbol{w}_{1:n},s_n|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)p\left(\boldsymbol{w}_{n+1:N}|s_n,\boldsymbol{\Pi},\boldsymbol{\phi}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)} \\
&= \frac{\mathcal{A}_n(s_n)\mathcal{B}_n(s_n)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}\right)} \\
&\propto \mathcal{A}_n(s_n)\mathcal{B}_n(s_n)
\end{aligned}$$

Derivation of eq. (*8.15*)

$$\begin{aligned}
\mathcal{B}_n(s_n) &= p\left(\boldsymbol{w}_{n+1:N}|s_n,\boldsymbol{\Pi},\boldsymbol{\phi}\right) \\
&= \sum_{s_{n+1}} p\left(\boldsymbol{w}_{n+1:N},s_{n+1}|s_n,\boldsymbol{\Pi},\boldsymbol{\phi}\right) \\
&= \sum_{s_{n+1}} p\left(\boldsymbol{w}_{n+2:N}|w_{n+1},s_{n+1},s_n,\boldsymbol{\Pi},\boldsymbol{\phi}\right)p\left(w_{n+1}|s_{n+1},s_n,\boldsymbol{\Pi},\boldsymbol{\phi}\right)p\left(s_{n+1}|s_n,\boldsymbol{\Pi},\boldsymbol{\phi}\right) \\
&= \sum_{s_{n+1}} p\left(\boldsymbol{w}_{n+2:N}|s_{n+1}\boldsymbol{\Pi},\boldsymbol{\phi}\right)p\left(w_{n+1}|s_{n+1},\boldsymbol{\phi}\right)p\left(s_{n+1}|s_n,\boldsymbol{\Pi}\right) \\
&= \sum_{s_{n+1}} \mathcal{B}_{n+1}(s_{n+1})G\left(w_{n+1};\phi_{s_{n+1}}\right)\pi_{s_n\to s_{n+1}}
\end{aligned}$$

Derivation of eq. (*8.17*)

$$\begin{aligned}
p(s_N|\boldsymbol{w}_{1:N},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}) &= \frac{p(\boldsymbol{w}_{1:N},s_N|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})}{p(\boldsymbol{w}_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})} \\
&= \frac{\mathcal{A}_N(s_N)}{p(\boldsymbol{w}_{1:N}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})} \\
&\propto \mathcal{A}_N(s_N)
\end{aligned}$$

Derivation of eq. (*8.18*)

$$\begin{aligned}
p(s_n|\boldsymbol{s}_{n+1:N},\boldsymbol{w}_{1:N},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}) &= p(s_n|s_{n+1},\boldsymbol{w}_{1:n},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}) \\
&= \frac{p(s_n,s_{n+1}|\boldsymbol{w}_{1:n},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})}{p(s_{n+1}|\boldsymbol{w}_{1:n},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})} \\
&= \frac{p(s_{n+1}|s_n,\boldsymbol{w}_{1:n},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})}{p(s_{n+1}|\boldsymbol{w}_{1:n},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})}p(s_n|\boldsymbol{w}_{1:n},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}) \\
&= \frac{p(s_{n+1}|s_n,\boldsymbol{w}_{1:n},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})}{p(s_{n+1}|\boldsymbol{w}_{1:n},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})}\frac{p(\boldsymbol{w}_{1:n},s_n|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})}{p(\boldsymbol{w}_{1:n}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})} \\
&= \frac{p(s_{n+1}|s_n,\boldsymbol{\Pi})}{p(s_{n+1}|\boldsymbol{w}_{1:n},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})}\frac{p(\boldsymbol{w}_{1:n},s_n|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})}{p(\boldsymbol{w}_{1:n}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})} \\
&= \frac{\pi_{s_n\to s_{n+1}}}{p(s_{n+1}|\boldsymbol{w}_{1:n},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})}\frac{\mathcal{A}_n(s_n)}{p(\boldsymbol{w}_{1:n}|\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi})} \\
&\propto \pi_{s_n\to s_{n+1}}\mathcal{A}_n(s_n)
\end{aligned}$$

E.0.3 Relations of section 8.3.3

Derivation of eq. (*8.19*)

$$\begin{aligned}
\log p\left(\boldsymbol{s}_{1:N}, \boldsymbol{w}_{1:N} | \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) &= \log p\left(\boldsymbol{s}_{1:N} | \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) + \log p\left(\boldsymbol{w}_{1:N} | \boldsymbol{s}_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) \\
&= \log p\left(\boldsymbol{s}_{1:N} | \boldsymbol{\rho}, \boldsymbol{\Pi}\right) + \log p\left(\boldsymbol{w}_{1:N} | \boldsymbol{s}_{1:N}, \boldsymbol{\phi}\right) \\
&= \log p\left(s_1 | \boldsymbol{\rho}\right) + \log \prod_{n=2}^{N} p\left(s_n | s_{n-1}, \boldsymbol{\Pi}\right) + \log \prod_{n=1}^{N} p\left(w_n | s_n, \boldsymbol{\phi}\right) \\
&= \log p\left(s_1 | \boldsymbol{\rho}\right) + \sum_{n=2}^{N} \log p\left(s_n | s_{n-1}, \boldsymbol{\Pi}\right) + \sum_{n=1}^{N} \log p\left(w_n | s_n, \boldsymbol{\phi}\right) \\
&= \log \rho_{s_1} + \sum_{n=2}^{N} \log \pi_{s_{n-1} \to s_n} + \sum_{n=1}^{N} \log G\left(w_n; \phi_{s_n}\right)
\end{aligned}$$

Derivation of eq. (*8.20*)

$$\begin{aligned}
Q_{\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}}(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) &= \sum_{\boldsymbol{s}_{1:N}} p\left(\boldsymbol{s}_{1:N} | \boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) \log p\left(\boldsymbol{s}_{1:N}, \boldsymbol{w}_{1:N} | \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) \\
&= \sum_{\boldsymbol{s}_{1:N}} p\left(\boldsymbol{s}_{1:N} | \boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) \log \rho_{s_1} \\
&+ \sum_{\boldsymbol{s}_{1:N}} p\left(\boldsymbol{s}_{1:N} | \boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) \sum_{n=2}^{N} \log \pi_{s_{n-1} \to s_n} \\
&+ \sum_{\boldsymbol{s}_{1:N}} p\left(\boldsymbol{s}_{1:N} | \boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) \sum_{n=1}^{N} \log G\left(w_n; \phi_{s_n}\right) \\
&= \sum_{s_1} p\left(s_1 | \boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) \log \rho_{s_1} \\
&+ \sum_{\boldsymbol{s}_{1:N}} \sum_{n=2}^{N} p\left(\boldsymbol{s}_{1:N} | \boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) \log \pi_{s_{n-1} \to s_n} \\
&+ \sum_{\boldsymbol{s}_{1:N}} \sum_{n=1}^{N} p\left(\boldsymbol{s}_{1:N} | \boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) \log G\left(w_n; \phi_{s_n}\right) \\
&= \sum_{s_1} p\left(s_1 | \boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) \log \rho_{s_1} \\
&+ \sum_{s_{n-1}} \sum_{n=2}^{N} \sum_{s_n} p\left(s_{n-1}, s_n | \boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) \log \pi_{s_{n-1} \to s_n} \\
&+ \sum_{s_n} \sum_{n=1}^{N} p\left(s_n | \boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) \log G\left(w_n; \phi_{s_n}\right)
\end{aligned}$$

Derivation of eq. (*8.21*)

$$p\left(s_n | \boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) = \frac{p\left(\boldsymbol{w}_{1:n}, s_n, \boldsymbol{w}_{n+1:N} | \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N} | \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}$$

$$
\begin{aligned}
&= \frac{p\left(\boldsymbol{w}_{1:n}, s_n|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) p\left(\boldsymbol{w}_{n+1:N}|\boldsymbol{w}_{1:n}, s_n, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)} \\
&= \frac{p\left(\boldsymbol{w}_{1:n}, s_n|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) p\left(\boldsymbol{w}_{n+1:N}|s_n, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)} \\
&= \frac{\mathcal{A}_n^{old}(s_n)\mathcal{B}_n^{old}(s_n)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}
\end{aligned}
$$

Derivation of eq. (*8.22*)

$$
\begin{aligned}
p\left(s_{n-1}, s_n|\boldsymbol{w}_{1:N}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) &= \frac{p\left(\boldsymbol{w}_{1:n}, \boldsymbol{w}_{n+1:N}, s_{n-1}, s_n|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)} \\
&= \frac{p\left(\boldsymbol{w}_{n+1:N}|\boldsymbol{w}_{1:n}, s_{n-1}, s_n, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) p\left(\boldsymbol{w}_{1:n}, s_{n-1}, s_n|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)} \\
&= \frac{p\left(\boldsymbol{w}_{n+1:N}|s_n, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) p\left(\boldsymbol{w}_{1:n}, s_{n-1}, s_n|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)} \\
&= \frac{\mathcal{B}_n^{old}(s_n) p\left(\boldsymbol{w}_{1:n}, s_{n-1}, s_n|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)} \\
&= \frac{\mathcal{B}_n^{old}(s_n) p\left(w_n|\boldsymbol{w}_{1:n-1}, s_{n-1}, s_n, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) p\left(\boldsymbol{w}_{1:n-1}, s_{n-1}, s_n|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)} \\
&= \frac{\mathcal{B}_n^{old}(s_n) p\left(w_n|s_n, \boldsymbol{\phi}^{old}\right) p\left(\boldsymbol{w}_{1:n-1}, s_{n-1}, s_n|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)} \\
&= \frac{\mathcal{B}_n^{old}(s_n) G\left(w_n; \phi_{s_n}^{old}\right) p\left(\boldsymbol{w}_{1:n-1}, s_{n-1}, s_n|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)} \\
&= \frac{\mathcal{B}_n^{old}(s_n) G\left(w_n; \phi_{s_n}^{old}\right) p\left(s_n|\boldsymbol{w}_{1:n-1}, s_{n-1}, \boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right) p\left(\boldsymbol{w}_{1:n-1}, s_{n-1}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)} \\
&= \frac{\mathcal{B}_n^{old}(s_n) G\left(w_n; \phi_{s_n}^{old}\right) p\left(s_n|s_{n-1}, \boldsymbol{\Pi}^{old}\right) p\left(\boldsymbol{w}_{1:n-1}, s_{n-1}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)} \\
&= \frac{\mathcal{B}_n^{old}(s_n) G\left(w_n; \phi_{s_n}^{old}\right) \pi_{s_{n-1}\rightarrow s_n}^{old} \mathcal{A}_{n-1}^{old}(s_{n-1})}{p\left(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}^{old}, \boldsymbol{\Pi}^{old}, \boldsymbol{\phi}^{old}\right)}
\end{aligned}
$$

Derivation of eq. (*8.26*)

The gradient of the Lagrangian is equal to

$$
\frac{\partial \mathcal{L}\left(\lambda, \rho_{\sigma_1}, \ldots, \rho_{\sigma_M}\right)}{\partial \lambda} = 1 - \sum_{\sigma_m} \rho_{\sigma_m}
$$

$$\frac{\partial \mathcal{L}\left(\lambda, \rho_{\sigma_1}, \ldots, \rho_{\sigma_M}\right)}{\partial \rho_{\sigma_m}} = \frac{\zeta_1^{old}(\sigma_m)}{\rho_{\sigma_m}} - \lambda, \qquad m = 1, \ldots, M$$

Solving $\partial\mathcal{L}/\partial\rho_{\sigma_m} = 0$ implies $\rho_{\sigma_m} = \zeta_1^{old}(\sigma_m)/\lambda$. Using $\partial\mathcal{L}/\partial\lambda = 0$ gives $\lambda = \sum_{\sigma_m} \zeta_1^{old}(\sigma_m)$.

Derivation of eq. (*8.27*)

The gradient of the Lagrangian is equal to

$$\frac{\partial \mathcal{K}_m\left(\kappa_m, \pi_{\sigma_m\to\sigma_1}, \ldots, \pi_{\sigma_m\to\sigma_M}\right)}{\partial \kappa_m} = 1 - \sum_{\sigma_{m'}} \pi_{\sigma_m\to\sigma_{m'}}$$

$$\frac{\partial \mathcal{K}_m\left(\kappa_m, \pi_{\sigma_m\to\sigma_1}, \ldots, \pi_{\sigma_m\to\sigma_M}\right)}{\partial \pi_{\sigma_m\to\sigma_{m'}}} = \frac{\sum_{n=2}^{N} \eta_n^{old}(\sigma_m, \sigma_{m'})}{\pi_{\sigma_m\to\sigma_{m'}}} - \kappa_m, \qquad m' = 1, \ldots, M$$

Solving $\partial\mathcal{K}_m/\partial\pi_{\sigma_m\to\sigma_{m'}} = 0$ implies $\pi_{\sigma_m\to\sigma_{m'}} = \sum_{n=2}^{N} \eta_n^{old}(\sigma_m, \sigma_{m'})/\kappa_m$. Using $\partial\mathcal{K}/\partial\kappa_m = 0$ gives $\kappa_m = \sum_{\sigma_{m'}} \sum_{n=2}^{N} \eta_n^{old}(\sigma_m, \sigma_{m'})$.

E.0.4 Relations of section 8.3.4

Derivation of eq. (*8.29*)

$$\begin{aligned}
\hat{\mathcal{A}}_n(s_n) &= p(s_n|\boldsymbol{w}_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= \frac{p(w_n, s_n|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})}{p(w_n|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} \\
&= \frac{p(w_n|s_n, \boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})}{p(w_n|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} p(s_n|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= \frac{p(w_n|s_n, \boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})}{p(w_n|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} \sum_{s_{n-1}} p(s_n, s_{n-1}|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= \frac{p(w_n|s_n, \boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})}{p(w_n|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} \sum_{s_{n-1}} p(s_n|s_{n-1}, \boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) p(s_{n-1}|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= \frac{p(w_n|s_n, \boldsymbol{\phi})}{p(w_n|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} \sum_{s_{n-1}} p(s_n|s_{n-1}, \boldsymbol{\Pi}) p(s_{n-1}|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\
&= \frac{G\left(w_n; \phi_{s_n}\right)}{p(w_n|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} \sum_{s_{n-1}} \pi_{s_{n-1}\to s_n} \hat{\mathcal{A}}_{n-1}(s_{n-1}) \\
&= \frac{1}{\hat{\mathcal{C}}_n} G\left(w_n; \phi_{s_n}\right) \sum_{s_{n-1}} \pi_{s_{n-1}\to s_n} \hat{\mathcal{A}}_{n-1}(s_{n-1})
\end{aligned}$$

Derivation of eq. (*8.30*)

$$\begin{aligned}
\hat{\mathcal{B}}_n(s_n) &= \frac{p\left(\boldsymbol{w}_{n+1:N}|s_n, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)}{p\left(\boldsymbol{w}_{n+1:N}|\boldsymbol{w}_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)} \\
&= \frac{\sum_{s_{n+1}} p\left(\boldsymbol{w}_{n+1:N}, s_{n+1}|s_n, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)}{p\left(\boldsymbol{w}_{n+2:N}|\boldsymbol{w}_{1:n+1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) p\left(w_{n+1}|\boldsymbol{w}_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)} \\
&= \frac{\sum_{s_{n+1}} p\left(w_{n+1}|\boldsymbol{w}_{n+2:N}, s_{n+1}, s_n, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) p\left(\boldsymbol{w}_{n+2:N}|s_{n+1}, s_n, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) p\left(s_{n+1}|s_n, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)}{p\left(\boldsymbol{w}_{n+2:N}|\boldsymbol{w}_{1:n+1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) p\left(w_{n+1}|\boldsymbol{w}_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)} \\
&= \frac{1}{p\left(w_{n+1}|\boldsymbol{w}_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)} \sum_{s_{n+1}} p\left(w_{n+1}|s_{n+1}, \boldsymbol{\phi}\right) \frac{p\left(\boldsymbol{w}_{n+2:N}|s_{n+1}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)}{p\left(\boldsymbol{w}_{n+2:N}|\boldsymbol{w}_{1:n+1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)} p\left(s_{n+1}|s_n, \boldsymbol{\Pi}\right) \\
&= \frac{1}{\hat{\mathcal{C}}_{n+1}} \sum_{s_{n+1}} G\left(w_{n+1}; \phi_{s_{n+1}}\right) \hat{\mathcal{B}}_{n+1}(s_{n+1}) \pi_{s_n\to s_{n+1}}
\end{aligned}$$

Derivation of eq. (*8.32*)

$$\begin{aligned}\hat{\mathcal{A}}_n(s_n) &= p(s_n|\boldsymbol{w}_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\ &= \frac{1}{p(\boldsymbol{w}_{1:n}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} p(\boldsymbol{w}_{1:n}, s_n|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\ &= \frac{1}{p(\boldsymbol{w}_{1:n}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} \mathcal{A}_n(s_n)\end{aligned}$$

Derivation of eq. (*8.33*)

$$\begin{aligned}\hat{\mathcal{B}}_n(s_n) &= \frac{1}{p\left(\boldsymbol{w}_{n+1:N}|\boldsymbol{w}_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)} p\left(\boldsymbol{w}_{n+1:N}|s_n, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) \\ &= \frac{1}{p\left(\boldsymbol{w}_{n+1:N}|\boldsymbol{w}_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)} \mathcal{B}_n(s_n)\end{aligned}$$

Derivation of eq. (*8.34*)

$$\begin{aligned}p(\boldsymbol{w}_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) &= p(w_1|\boldsymbol{\rho}, \boldsymbol{\phi}) \prod_{n=2}^{N} p(w_n|\boldsymbol{w}_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \\ &= \prod_{n=1}^{N} \hat{\mathcal{C}}_n\end{aligned}$$

E.0.5 Relations of section 8.4.2

Derivation of eq. (*8.42*)

$$p\left(s_N|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:N}\right) = \hat{\mathcal{A}}_n(s_n)$$

as readily implied by the definition of $\hat{\mathcal{A}}_N(s_N)$.
Derivation of eq. (*8.43*)

$$\begin{aligned}p\left(s_n|\boldsymbol{s}_{n+1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:N}\right) &= p\left(s_n|s_{n+1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:n}\right) \\ &= \frac{p\left(s_n, s_{n+1}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:n}\right)}{p\left(s_{n+1}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:n}\right)} \\ &= \frac{p\left(s_{n+1}|s_n, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:n}\right) p\left(s_n|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:n}\right)}{p\left(s_{n+1}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:n}\right)} \\ &= \frac{p\left(s_{n+1}|s_n, \boldsymbol{\Pi}\right) p\left(s_n|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:n}\right)}{p\left(s_{n+1}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:n}\right)} \\ &= \frac{\pi_{s_n \to s_{n+1}} \mathcal{A}_n(s_n)}{p\left(s_{n+1}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:n}\right)} \\ &\propto \pi_{s_n \to s_{n+1}} \mathcal{A}_n(s_n)\end{aligned}$$

Although unnecessary, normalization can be recovered by $\sum_{s_n} p\left(s_n|\boldsymbol{s}_{n+1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, \boldsymbol{w}_{1:N}\right) = 1$.
Derivation of eq. (*8.44*) TODO

$$\boldsymbol{\rho}^{new} \sim \text{Dirichlet}_M\left(\eta\boldsymbol{\zeta} + \boldsymbol{d}^{new}\right)$$

Derivation of eq. (*8.45*) TODO

$$\boldsymbol{\pi}_{\sigma_m}^{new} \sim \text{Dirichlet}_M\left(\alpha_{\sigma_m}\boldsymbol{\beta}_{\sigma_m} + \boldsymbol{c}_{\sigma_m}^{new}\right)$$

Derivation of eq. (*8.46*) TODO

$$p(\phi_{\sigma_m}|\boldsymbol{s}_{1:N}^{new}, \boldsymbol{w}_{1:N}) \propto H(\phi_{\sigma_m}) \prod_{n \in \mathcal{N}_{\sigma_m}^{new}} G(w_n; \phi_{\sigma_m})$$

E.0.6 Relations of section 8.6

Derivation of **??**

$$
\begin{aligned}
p\left(\boldsymbol{\beta}|\boldsymbol{s}_{1:N}\right) &= \int_{\boldsymbol{\rho}} d\boldsymbol{\rho} \int_{\boldsymbol{\pi}_{\sigma_1}} d\boldsymbol{\pi}_{\sigma_1} \cdots \int_{\boldsymbol{\pi}_{\sigma_M}} d\boldsymbol{\pi}_{\sigma_M} p\left(\boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\pi}_{\sigma_1}, \cdots, \boldsymbol{\pi}_{\sigma_M}|\boldsymbol{s}_{1:N}\right) \\
&= \int_{\boldsymbol{\rho}} d\boldsymbol{\rho} \int_{\boldsymbol{\pi}_{\sigma_1}} d\boldsymbol{\pi}_{\sigma_1} \cdots \int_{\boldsymbol{\pi}_{\sigma_M}} d\boldsymbol{\pi}_{\sigma_M} \frac{p\left(\boldsymbol{s}_{1:N}|\boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\pi}_{\sigma_1}, \cdots, \boldsymbol{\pi}_{\sigma_M}\right) p\left(\boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\pi}_{\sigma_1}, \cdots, \boldsymbol{\pi}_{\sigma_M}\right)}{p(\boldsymbol{s}_{1:N})} \\
&= \int_{\boldsymbol{\rho}} d\boldsymbol{\rho} \int_{\boldsymbol{\pi}_{\sigma_1}} d\boldsymbol{\pi}_{\sigma_1} \cdots \int_{\boldsymbol{\pi}_{\sigma_M}} d\boldsymbol{\pi}_{\sigma_M} \frac{p\left(\boldsymbol{s}_{1:N}|\boldsymbol{\rho}, \boldsymbol{\pi}_{\sigma_1}, \cdots, \boldsymbol{\pi}_{\sigma_M}\right) p\left(\boldsymbol{\rho}, \boldsymbol{\pi}_{\sigma_1}, \cdots, \boldsymbol{\pi}_{\sigma_M}|\boldsymbol{\beta}\right)}{p(\boldsymbol{s}_{1:N})} p\left(\boldsymbol{\beta}\right) \\
&= \int_{\boldsymbol{\rho}} d\boldsymbol{\rho} \int_{\boldsymbol{\pi}_{\sigma_1}} d\boldsymbol{\pi}_{\sigma_1} \cdots \int_{\boldsymbol{\pi}_{\sigma_M}} d\boldsymbol{\pi}_{\sigma_M} \frac{\prod_{\sigma_m} \rho_{\sigma_m}^{d_{\sigma_m}} \times \prod_{\sigma_m} \prod_{\sigma_{m'}} \pi_{\sigma_m \to \sigma_{m'}}^{c_{\sigma_m \to \sigma_{m'}}} \times p\left(\boldsymbol{\rho}|\boldsymbol{\beta}\right) \times \prod_{\sigma_m} p\left(\boldsymbol{\pi}_{\sigma_m}|\boldsymbol{\beta}\right)}{p(\boldsymbol{s}_{1:N})} p\left(\boldsymbol{\beta}\right) \\
&= \int_{\boldsymbol{\rho}} d\boldsymbol{\rho} \int_{\boldsymbol{\pi}_{\sigma_1}} d\boldsymbol{\pi}_{\sigma_1} \cdots \int_{\boldsymbol{\pi}_{\sigma_M}} d\boldsymbol{\pi}_{\sigma_M} \frac{\prod_{\sigma_m} \rho_{\sigma_m}^{\alpha\beta_{\sigma_m} + d_{\sigma_m} - 1} \times \prod_{\sigma_m} \prod_{\sigma_{m'}} \pi_{\sigma_m \to \sigma_{m'}}^{\alpha\beta_{\sigma_{m'}} + c_{\sigma_m \to \sigma_{m'}} - 1} \times \left(\frac{\Gamma(\alpha)}{\prod_{\sigma_m} \Gamma(\alpha\beta_{\sigma_m})}\right)^{M+1}}{p(\boldsymbol{s}_{1:N})} p\left(\boldsymbol{\beta}\right) \\
&= \int_{\boldsymbol{\rho}} d\boldsymbol{\rho} \int_{\boldsymbol{\pi}_{\sigma_1}} d\boldsymbol{\pi}_{\sigma_1} \cdots \int_{\boldsymbol{\pi}_{\sigma_M}} d\boldsymbol{\pi}_{\sigma_M} \frac{\frac{\prod_{\sigma_m} \Gamma(\alpha\beta_{\sigma_m} + d_{\sigma_m})}{\Gamma(\alpha + D)} \text{Dirichlet}_M(\boldsymbol{\rho}; \alpha\boldsymbol{\beta} + \boldsymbol{d}) \times \prod_{\sigma_m} \frac{\prod_{\sigma_{m'}} \Gamma(\alpha\beta_{\sigma_{m'}} + c_{\sigma_m \to \sigma_{m'}})}{\Gamma(\alpha + C_{\sigma_m})} \text{Dirichlet}}{p(\boldsymbol{s}_{1:N})} \\
&= \frac{\frac{\prod_{\sigma_m} \Gamma(\alpha\beta_{\sigma_m} + d_{\sigma_m})}{\Gamma(\alpha + D)} \times \prod_{\sigma_m} \frac{\prod_{\sigma_{m'}} \Gamma(\alpha\beta_{\sigma_{m'}} + c_{\sigma_m \to \sigma_{m'}})}{\Gamma(\alpha + C_{\sigma_m})} \times \left(\frac{\Gamma(\alpha)}{\prod_{\sigma_m} \Gamma(\alpha\beta_{\sigma_m})}\right)^{M+1}}{p(\boldsymbol{s}_{1:N})} p\left(\boldsymbol{\beta}\right) \\
&= \frac{\Gamma(\alpha)}{\Gamma(\alpha + D)} \prod_{\sigma_m} \frac{\Gamma(\alpha\beta_{\sigma_m} + d_{\sigma_m})}{\Gamma(\alpha\beta_{\sigma_m})} \times \prod_{\sigma_m} \frac{\Gamma(\alpha)}{\Gamma(\alpha + C_{\sigma_m})} \prod_{\sigma_{m'}} \frac{\Gamma(\alpha\beta_{\sigma_{m'}} + c_{\sigma_m \to \sigma_{m'}})}{\Gamma(\alpha\beta_{\sigma_{m'}})} \times \frac{p\left(\boldsymbol{\beta}\right)}{p(\boldsymbol{s}_{1:N})} \\
&= \frac{\Gamma(\alpha)}{\Gamma(\alpha + D)} \prod_{\sigma_m} \frac{\Gamma(\alpha\beta_{\sigma_m} + d_{\sigma_m})}{\Gamma(\alpha\beta_{\sigma_m})} \times \prod_{\sigma_m} \frac{\Gamma(\alpha)}{\Gamma(\alpha + C_{\sigma_m})} \prod_{\sigma_{m'}} \frac{\Gamma(\alpha\beta_{\sigma_{m'}} + c_{\sigma_m \to \sigma_{m'}})}{\Gamma(\alpha\beta_{\sigma_{m'}})} \times \frac{\Gamma(\gamma)}{\left(\Gamma(\gamma/M)\right)^M} \frac{\prod_{\sigma_m} \beta_{\sigma_m}^{\frac{\gamma}{M}-1}}{p(\boldsymbol{s}_{1:N})} \\
&\propto \prod_{\sigma_m} \frac{\Gamma(\alpha\beta_{\sigma_m} + d_{\sigma_m})}{\Gamma(\alpha\beta_{\sigma_m})} \times \prod_{\sigma_m} \prod_{\sigma_{m'}} \frac{\Gamma(\alpha\beta_{\sigma_{m'}} + c_{\sigma_m \to \sigma_{m'}})}{\Gamma(\alpha\beta_{\sigma_{m'}})} \times \prod_{\sigma_m} \beta_{\sigma_m}^{\frac{\gamma}{M}-1} \\
&= \prod_{\sigma_m} \left(\beta_{\sigma_m}^{\frac{\gamma}{M}-1} \times \frac{\Gamma(\alpha\beta_{\sigma_m} + d_{\sigma_m})}{\Gamma(\alpha\beta_{\sigma_m})} \times \prod_{\sigma_{m'}} \frac{\Gamma(\alpha\beta_{\sigma_{m'}} + c_{\sigma_m \to \sigma_{m'}})}{\Gamma(\alpha\beta_{\sigma_{m'}})} \right)
\end{aligned}
$$

where the counts $\boldsymbol{d}$ and $\boldsymbol{c}_{\sigma_m}$ are similar to eqs. (*8.44*) and (*8.45*); while, $D = \sum_{\sigma_m} d_{\sigma_m}$ and $C_{\sigma_m} = \sum_{\sigma_{m'}} c_{\sigma_m \to \sigma_{m'}}$.

Bibliography