Structured Probabilistic Reasoning

(Incomplete draft)

Bart Jacobs
Institute for Computing and Information Sciences, Radboud University Nijmegen,
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands.
bart@cs.ru.nl http://www.cs.ru.nl/~bart

Version of: August 4, 2019
**Contents**

<table>
<thead>
<tr>
<th>Preface</th>
<th>page v</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Collections and Channels</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Cartesian products</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Lists</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Powersets</td>
<td>9</td>
</tr>
<tr>
<td>1.4 Multisets</td>
<td>13</td>
</tr>
<tr>
<td>1.5 Probability distributions</td>
<td>22</td>
</tr>
<tr>
<td>1.6 Frequentist learning: from multisets to distributions</td>
<td>29</td>
</tr>
<tr>
<td>1.7 Channels</td>
<td>33</td>
</tr>
<tr>
<td>1.8 Parallel products</td>
<td>40</td>
</tr>
<tr>
<td>1.9 A Bayesian network example</td>
<td>50</td>
</tr>
<tr>
<td>1.10 The role of category theory</td>
<td>56</td>
</tr>
<tr>
<td><strong>2 Predicates and Observables</strong></td>
<td>62</td>
</tr>
<tr>
<td>2.1 Validity</td>
<td>63</td>
</tr>
<tr>
<td>2.2 The structure of observables</td>
<td>71</td>
</tr>
<tr>
<td>2.3 Conditioning</td>
<td>79</td>
</tr>
<tr>
<td>2.4 Transformation of observables</td>
<td>88</td>
</tr>
<tr>
<td>2.5 Reasoning along channels</td>
<td>95</td>
</tr>
<tr>
<td>2.6 Discretisation, and coin bias learning</td>
<td>104</td>
</tr>
<tr>
<td>2.7 Inference in Bayesian networks</td>
<td>108</td>
</tr>
<tr>
<td>2.8 Validity-based distances</td>
<td>112</td>
</tr>
<tr>
<td>2.9 Variance, covariance and correlation</td>
<td>119</td>
</tr>
<tr>
<td>2.10 Dependence and covariance</td>
<td>126</td>
</tr>
<tr>
<td><strong>3 Directed Graphical Models</strong></td>
<td>131</td>
</tr>
<tr>
<td>3.1 String diagrams</td>
<td>133</td>
</tr>
<tr>
<td>3.2 Equations for string diagrams</td>
<td>141</td>
</tr>
<tr>
<td>3.3 Accessibility and joint states</td>
<td>146</td>
</tr>
<tr>
<td>Section Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.4 Hidden Markov models</td>
<td>150</td>
</tr>
<tr>
<td>3.5 Disintegration</td>
<td>161</td>
</tr>
<tr>
<td>3.6 Disintegration for states</td>
<td>167</td>
</tr>
<tr>
<td>3.7 Bayesian inversion</td>
<td>170</td>
</tr>
<tr>
<td>3.8 Categorical aspects of Bayesian inversion</td>
<td>179</td>
</tr>
<tr>
<td>3.9 Factorisation of joint states</td>
<td>186</td>
</tr>
<tr>
<td>3.10 Inference in Bayesian networks, reconsidered</td>
<td>194</td>
</tr>
<tr>
<td>3.11 Updating and adapting: Pearl versus Jeffrey</td>
<td>199</td>
</tr>
<tr>
<td><strong>4 Learning of States and Channels</strong></td>
<td>210</td>
</tr>
<tr>
<td>4.1 Learning a channel from tabular data</td>
<td>211</td>
</tr>
<tr>
<td>4.2 Learning with missing data</td>
<td>214</td>
</tr>
<tr>
<td>4.3 Increasing validity via updating</td>
<td>219</td>
</tr>
<tr>
<td>4.4 A general sum-increase method</td>
<td>222</td>
</tr>
<tr>
<td>4.5 Learning Markov chains</td>
<td>229</td>
</tr>
<tr>
<td>4.6 Learning hidden Markov models</td>
<td>234</td>
</tr>
<tr>
<td>4.7 Data, validity, and learning</td>
<td>241</td>
</tr>
<tr>
<td>4.8 Learning along a channel</td>
<td>250</td>
</tr>
<tr>
<td>4.9 Expectation Maximisation</td>
<td>260</td>
</tr>
<tr>
<td>4.10 A clustering example</td>
<td>273</td>
</tr>
<tr>
<td><strong>References</strong></td>
<td>279</td>
</tr>
</tbody>
</table>
The phrase ‘structure in probability’ sounds like a *contradictio in terminis*: it seems that probability is about randomness, like in the tossing of coins, in which one may not expect to find much structure. Still, as we know since the seventeenth century, via the pioneering work of Christiaan Huygens, Pierre Fermat, and Blaise Pascal, there is quite some mathematical structure in the area of probability. The *raison d’être* of this book is that there is more structure — especially algebraic and categorical — than is commonly emphasised.

The scientific roots of this book’s author lie outside probability theory, in type theory and logic (including some quantum logic), in semantics and specification of programming languages, in computer security and privacy, in state-based computation (coalgebra), and in category theory. This scientific distance to probability theory has advantages and disadvantages. Its obvious disadvantage is that there is no deeply engrained familiarity with the field and with its development. But at the same time this distance may be an advantage, since it provides a fresh perspective, without sacred truths and without adherence to common practices and notations. For instance, the terminology and notation in this book are somewhat influenced by quantum theory, for instance in using the word ‘state’ for ‘distribution’ or the word ‘observable’ for an $\mathbb{R}$-valued function on a sample space, in using ket notation $| - \rangle$ for discrete probability distributions, or in using daggers as reversals in analogy with conjugate transposes (for Hilbert spaces).

It should be said: for someone trained in formal methods, the area of probability theory can be rather sloppy: everything is called ‘$P$’, types are hardly ever used, crucial ingredients (like distributions in expected values) are left implicit, basic notions (like conjugate prior) are introduced only via examples, calculation recipes and algorithms are regularly just given, without explanation, goal or justification, *etc.* This hurts, especially because there is so much beautiful mathematical structure around. For instance, the notion of a channel
formalises the idea of a conditional probability and carries a rich mathematical structure that can be used in compositional reasoning, with both sequential and parallel composition. The Bayesian inversion (‘dagger’) of a channel does not only come with appealing mathematical (categorical) properties — e.g. smooth interaction with sequential and parallel composition — but is also extremely useful in inference (in Jeffrey’s rule) and in connecting forward and backward inference via reversal (see Theorem 3.7.7, the author’s favourite), and in learning. It is a real pity that such pearls of the field receive relatively little attention.

We even dare to think that this ‘sloppiness’ is ultimately a hindrance to further development of the field, especially in computer science. For instance, it is hard to even express a result like Theorem 3.7.7 in standard probabilistic notation. One can speculate that state/distributions are kept implicit in traditional probability theory because in many examples they are used as a fixed implicit assumption in the background. Indeed, in mathematical notation one tends to omit — for efficiency — the least relevant (implicit) parameters. But the essence of probabilistic computation is state transformation, where it has become highly relevant to write explicitly in which state one is working at which moment. Similarly, the difference between a ‘multiple-state’ and a ‘copied-state’ perspective — which plays an important role in this book, e.g. in learning — comes naturally once states are made explicit. The notation developed in this book helps in such situations — and in many other situations as well, we hope.

This book uses the notion of a (probabilistic) channel as cornerstone of its ‘channel-based’ account of probabilistic reasoning. It does so while emphasising the use of channels (or: Kleisli maps, in categorical terms) for other forms of computation. Indeed, the book starts by laying out the similarities between the basic ‘collection’ data types of lists, subsets, multisets, and (probability) distributions, with their own properties and associated channels, for instance for non-deterministic computation. Especially the interplay between multisets and distributions, notably in learning, is a recurring theme.

But it is not just mathematical aesthetics that drives the developments in this book. Probability theory nowadays forms the basis of large parts of big data analytics and of artificial intelligence. These areas are of increasing societal relevance and provide the basis of the modern view of the world — more based on correlation than on causation — and also provide the basis for much of modern decision making, that may affect the lives of billions of people in profound ways. There are increasing demands for justification of such probabilistic reasoning methods and decisions, for instance in the legal setting provided by Europe’s General Data Protection Regulation (GDPR). Its recital 71 is about automated decision-making and talks about a right to obtain an explanation:
In any case, such processing should be subject to suitable safeguards, which should include specific information to the data subject and the right to obtain human intervention, to express his or her point of view, to obtain an explanation of the decision reached after such assessment and to challenge the decision.

These and other developments have led to a new area called Explainable Artificial Intelligence (XAI), which strives to provide decisions with explanations that can be understood easily by humans, without bias or discrimination. Although this book will not contribute to XAI as such, it aims to provide a mathematically solid basis for such explanations.

In this context it is appropriate to quote Judea Pearl [81] from 1989 about a divide that is still wide today.

To those trained in traditional logics, symbolic reasoning is the standard, and non-monotonicity a novelty. To students of probability, on the other hand, it is symbolic reasoning that is novel, not nonmonotonicity. Dealing with new facts that cause probabilities to change abruptly from very high values to very low values is a commonplace phenomenon in almost every probabilistic exercise and, naturally, has attracted special attention among probabilists. The new challenge for probabilists is to find ways of abstracting out the numerical character of high and low probabilities, and cast them in linguistic terms that reflect the natural process of accepting and retracting beliefs.

This book does not pretend to fill this gap. One of the big embarrassments of the field is that there is no widely accepted symbolic logic for probability, together with proof rules and a denotational semantics. Such a logic will be a non-trivial, because it will have to be non-monotonic—one property that many logicians shy away from. This book does aim to contribute towards bridging the divide mentioned by Pearl, by providing a mathematical basis for such a symbolic probabilistic logic, consisting of channels, states, predicates, transformations, conditioning, disintegration, etc.

From the perspective of this book, the structured categorical approach to probability theory began with the work of Bill Lawvere (already in the 1960s) and his student Michèle Giry. They recognised that taking probability distributions has the structure of a monad, which was published in the early 1980s in [33]. Roughly at the same time Dexter Kozen started the systematic investigation of probabilistic programming languages and logics, published in [62, 63]. The monad introduced back then is now called the Giry monad $\mathcal{G}$, whose restriction to finite discrete probability distributions is written as $\mathcal{D}$. Much of this book, certainly in the beginning, concentrates on this discrete form. The language and notation that is used, however, covers both discrete

---

1 Informally, a logic is non-monotonic if adding assumptions may make a conclusion less true. For instance, I may think that scientists are civilised people, until, at some conference dinner, a heated scientific debate ends in a fist fight.
Chapter 0. Preface

and continuous probability — and quantum probability too (inspired by the general categorical notion of effectus, see [14, 41]).

Since the early 1980s the area remained relatively calm. It is only in the new millennium that there is renewed attention, sparked in particular by several developments.

- The grown interest in probabilistic programming languages that incorporate updating (conditioning) and/or higher order features, see e.g. [18, 19, 78, 95].
- The compositional approach to Bayesian networks [17, 29] and to Bayesian reasoning [53, 55].
- The use of categorical and diagrammatic methods in quantum foundations, including quantum probability, see [16] for an overview.

This book builds on these developments.

The intended audience consists of students and professionals — in mathematics, computer science, artificial intelligence and related fields — with a basic background in probability and in algebra and logic — and with an interest in formal, logically oriented approaches. This book’s goal is not to provide intuitive explanations of probability, like [97], but to provide clear and precise formalisations. Mathematical abstraction (esp. categorical air guitar playing) is not a goal in itself: instead, the book tries to uncover relevant abstractions in concrete problems. It includes several basic algorithms, with a focus on the algorithms’ correctness, not their efficiency. Each section ends with a series of exercises, so that the book can also be used for teaching and/or self-study. It aims at an undergraduate level. No familiarity with category theory is assumed. The basic, necessary notions are explained along the way. People who wish to learn more about category theory can use the references in the text, consult modern introductory texts like [4, 65], or use online resources such as ncatlab.org or Wikipedia.

The first chapter of the book covers introductory material that is meant to set the scene. It starts from basic collection types like lists and powersets, and continues with multisets and distributions, which will receive much attention. The chapter introduces the concept of a channel and shows how channels can be used for state transformation and how channels can be composed, both sequentially and in parallel. This culminates in an illustration of Bayesian networks in terms of (probabilistic) channels. At the same time it is shown how predictions are made about such Bayesian networks via state transformation and via compositional reasoning, basically by translating the network structure into (sequential and parallel) composites of channels.

The second chapter is more logically oriented, via observables (including
factors, predicates and events) that can be defined on sample spaces, providing numerical information. The chapter’s three key notions are validity, conditioning, and transformation of observables. The validity, or expected value, $\omega \models p$ of an observable $p$ in a state (distribution) $\omega$ captures the level of truth. Conditioning $\omega | p$ allows us to update a state with evidence given by $p$. It satisfied several basic properties, including Bayes’ rule. Where the first chapter introduced state transformation along a channel in a forward direction, the second chapter adds observable (predicate) transformation in a backward direction. These two operations are of fundamental importance in program semantics, and also in quantum computation — where they are distinguished as Schrödinger’s (forward) Heisenberg’s (backward) approach, see [45]. With these three basic notions in place, the techniques of forward inference and backward inference can be defined in a precise manner, via suitable combinations of conditioning and transformation. These inference techniques are illustrated in many examples, including Bayesian networks. The chapter ends with some further illustrations of the use of validity, in uniformly defining distances between states and between predicates, and in precisely defining (co)variance and correlation.

The third chapter is on (directed) graphical techniques. It first introduces string diagrams, which form a graphical language for channels — or, more abstractly, for symmetric monoidal categories. These string diagrams are similar to the graphs used for Bayesian networks, but they have explicit operations for copying and discarding and are thus more expressive. But most importantly, string diagrams have a clear semantics, namely in terms of channels. The chapter illustrates these string diagrams in a channel-based description of the basics of Markov chains and of hidden Markov models. But the most fundamental technique that is introduced in this chapter, via string diagrams, is disintegration. In essence, it is the well-known procedure of extracting a conditional probability $p(y \mid x)$ from a joint probability $p(x, y)$. One of the key features of the area of probability is that the components of a joint state ‘listen’ to each other, in the sense that if you update in one component, you see a change in another component — unless the joint is state is not entwined. One of the themes running through this book is how such ‘crossover’ influence can be captured via channels — extracted from joint states via disintegration — in particular via forward and backward inference. This phenomenon is what makes (reasoning in) Bayesian networks work. Disintegration is of interest in itself, but also leads to the Bayesian inversion of a channel. It gives a technique for reversing the direction of a channel. This corresponds to turning a conditional probability $p(y \mid x)$ into $p(x \mid y)$, essentially via Bayes’ rule. This Bayesian inversion is also called the ‘dagger’ of a channel, since it satisfies, as will be shown, the properties of a dagger operation (or conjugate transpose),
in Hilbert spaces (and in quantum theory). This dagger operation is actually of logical interest too, since it captures an alternative form of inference along a channel, commonly attributed to Richard Jeffrey. When precisely to use which inference rule is an open question.

Essentially all of the material in these first three chapters is known from the literature, but typically not in the channel-based form in which it is presented here. This book includes many examples, often copied from familiar sources, with the deliberate aim of illustrating how the channel-based approach actually works. Since many of these examples are taken from the literature, the interested reader may wish to compare the channel-based description used here with the original description.

The fourth chapter on learning is mostly new work, of which only a small part has been published so far (in [48]). One way of learning is to ask experts. However, it is more efficient and cheaper to learn from data, if available. This forms the basis of the whole data-driven economy that it now developing. One commonly distinguishes structure learning from parameter learning, where the former is about learning graphical structure in the data. This chapter is about parameter learning, for states and channels, based on what is called Maximal Likelihood Estimation (MLE). In essence, ‘learning’ is seen in this chapter as increasing a validity $\omega \models p$, by adjusting the state $\omega$ while keeping the evidence/data $p$ fixed. Hence, learning is about adjusting one’s state to the evidence at hand. The learning techniques used in this chapter are all based on a basic sum-increase lemma from [7]. It is shown how this technique can be used to learn the initial state and channel(s) for Markov chains and hidden Markov models. Such learning is known from the literature, but is described in here in channel-based form. This fourth chapter distinguishes data in various forms — such as multisets of points or multisets of factors — and uses different forms of validity, each with their own form of learning. In this way one can recognise different forms of learning, which in the literature are all labeled as Expectation-Maximisation (EM). This EM technique is explored here from a channel-based perspective, and is illustrated in many ways.

**Status of the current incomplete version**

An incomplete version of this book is made available online, in order to generate feedback and to justify a pause in the writing process. Feedback is most welcome, both positive and negative, especially when it suggests concrete improvements of the text. This may lead to occasional updates of this text. The date on the title page indicates the current version.

Some additional points.
• The (non-trivial) calculations in this book have been carried out with the EfProb library [12] for channel-based probability. When this book reaches its final form, the sources for these calculations will be made public. Also, several calculations in this book are (or: can be) done by hand, typically when the outcomes are described as fractions, like \(\frac{117}{2012}\). Such calculations are meant to be reconstructable by a motivated reader who really wishes to learn the ‘mechanics’ of the field. Outcomes obtained via EfProb are typically written as decimal notation 0.1234, as approximations, or as plots. They serve to give an impression of the results of a computation.

• The plans for the rest of this book, beyond Chapter 4, are not completely fixed yet, but will most likely involve additional chapters on: undirected graphical models, with (reversible) multiset-channels as underlying semantics, probabilistic programming and sampling, continuous probability, and possibly also a bit of causality and quantum probability.

There are several ways to combine elements from a given set, for instance as lists, subsets, multisets, and as probability distributions. This introductory chapter takes a systematic look at such collections and seeks to bring out many similarities. For instance, lists, subsets and multisets all form a monoid, essentially by suitable unions of collections. Unions of distributions are more subtle and take the form of convex combinations. Also, subsets, multisets and distributions can be combined naturally as parallel products $\otimes$, but not lists.

These collections are important in themselves, in many ways, but also as outputs of channels. Channels are functions of the form $\text{input} \rightarrow T(\text{output})$, where $T$ is a ‘collection’ operator, for instance, for lists, subsets, multisets, or distributions. Such channels capture a form of computation, directly related to the form of collection that is produced on the outputs. For instance, channels where $T$ is powerset are used as interpretations of non-deterministic computations, where each input element produces a subset of possible output elements. In the probabilistic case these channels produce distributions, for a different instantiation of the operator $T$. Channels can be used as elementary units of computation, which can be used to build more complicated computations via sequential and parallel composition.

Eventually, the chapter will concentrate on distributions, especially in Section 1.9 where an example of a Bayesian network is analysed. It is shown that the conditional probability tables that are associated with nodes in a Bayesian network are instances of probabilistic channels. As a result, one can systematically organise computations in Bayesian networks as suitable (sequential and/or parallel) compositions of channels. This is illustrated via calculation of predicted probabilities.

First, the reader is exposed to some general considerations that set the scene for what is coming. This requires some level of patience. But it is useful to see the similarities between distributions and other collections first, so that
constructions, techniques, notation, terminology and intuitions that we use for distributions can be put in a wider perspective and thus may become natural.

The final section of this chapter explains where the abstractions that we use come from, namely from category theory. It gives a quick overview of the most relevant parts of this theory and also how category theory will be used in the remainder of this book to elicit structure in probability.

1.1 Cartesian products

This section briefly reviews some (standard) terminology and notation related to Cartesian products of sets.

Let $X_1$ and $X_2$ be two arbitrary sets. We can form their Cartesian product $X_1 \times X_2$, as the new set containing all pairs of elements from $X_1$ and $X_2$, as in:

$$X_1 \times X_2 := \{(x_1, x_2) \mid x_1 \in X_1 \text{ and } x_2 \in X_2\}.$$ 

(We use the sign $\vdash$ for mathematical definitions.)

We thus write $(x_1, x_2)$ for the ‘pair’ or ‘tuple’ of elements $x_1 \in X_1$ and $x_2 \in X_2$. We have just defined a binary product set, constructed from two given sets $X_1, X_2$. We can also do this in $n$-ary form, for $n$ sets $X_1, \ldots, X_n$. We then get an $n$-ary Cartesian product:

$$X_1 \times \cdots \times X_n := \{(x_1, \ldots, x_n) \mid x_1 \in X_1 \text{ and } \cdots \text{ and } x_n \in X_n\}.$$

The tuple $(x_1, \ldots, x_n)$ is sometimes called an $n$-tuple. For convenience, it may be abbreviated as a vector $\vec{x}$. The product $X_1 \times \cdots \times X_n$ is sometimes written differently using the sign $\prod$, as:

$$\prod_{1 \leq i \leq n} X_i \quad \text{or more informally as:} \quad \prod X_i.$$

In the latter case it is left implicit what the range is of the index element $i$.

We allow $n = 0$. The resulting ‘empty’ product is then written as a singleton set 1, containing the empty tuple () as sole element, as in:

$$1 := \{()\}.$$

For $n = 1$ the product $X_1 \times \cdots \times X_n$ is (isomorphic to) the set $X_1$.

Notice that if one of the sets $X_i$ in a product $X_1 \times \cdots \times X_n$ is empty, then the whole product is empty. Also, if all of the sets $X_i$ are finite, then so is the product $X_1 \times \cdots \times X_n$. In fact, the number of element of $X_1 \times \cdots \times X_n$ is then obtained by multiplying all the numbers of elements of the sets $X_i$. 
1.1 Cartesian products

1.1.1 Projections and tuples

If we have sets $X_1, \ldots, X_n$ as above, then for each number $i$ with $1 \leq i \leq n$ there is a projection function $\pi_i$ out of the product to the set $X_i$, as in:

$$X_1 \times \cdots \times X_n \xrightarrow{\pi_i} X_i$$
given by $\pi_i(x_1, \ldots, x_n) := x_i$.

This gives us functions out of a product. We also wish to be able to define functions into a product, via tuples of functions: if we have a set $Y$ and $n$ functions $f_1: Y \to X_1$, $\ldots$, $f_n: Y \to X_n$, then we can form a new function $Y \to X_1 \times \cdots \times X_n$, namely:

$$Y \xrightarrow{(f_1, \ldots, f_n)} X_1 \times \cdots \times X_n \text{ via } (f_1, \ldots, f_n)(y) := (f_1(y), \ldots, f_n(y)).$$

There is an obvious result about projecting after tupling of functions:

$$\pi_i \circ (f_1, \ldots, f_n) = f_i.$$  \hspace{1cm} (1.1)

This is an equality of functions. It can be proven easily by applying both sides to an arbitrary element $y \in Y$.

There are some more ‘obvious’ equations about tupling of functions:

$$(f_1, \ldots, f_n) \circ g = (f_1 \circ g, \ldots, f_n \circ g) \quad \langle \pi_1, \ldots, \pi_n \rangle = \text{id}. \hspace{1cm} (1.2)$$

The latter function id is the identity function on the product $X_1 \times \cdots \times X_n$.

In a Cartesian product we put sets ‘in parallel’. We can also put functions between them in parallel. Suppose we have $n$ functions $f_i: X_i \to Y_i$. Then we can form the parallel composition:

$$X_1 \times \cdots \times X_n \xrightarrow{f_1 \times \cdots \times f_n} Y_1 \times \cdots \times Y_n$$

via:

$$f_1 \times \cdots \times f_n = (f_1 \circ \pi_1, \ldots, f_n \circ \pi_n)$$

so that:

$$(f_1 \times \cdots \times f_n)(x_1, \ldots, x_n) = (f_1(x_1), \ldots, f_n(x_n)).$$

The latter formulation clearly shows how the functions $f_i$ are applied in parallel to the elements $x_i$.

We overload the product symbol $\times$, since we use it both for sets and for functions. This may be a bit confusing at first, but it is in fact quite convenient.
1.1.2 Powers

Let $X$ be an arbitrary set. A power of $X$ is an $n$-product of $X$‘s, for some $n$. We write the $n$-th power of $X$ as $X^n$, in:

$$X^n := X \times \cdots \times X = \{ (x_1, \ldots, x_n) \mid x_i \in X \text{ for each } i \}.$$ 

As special cases we have $X^1 = X$ and $X^0 = 1$, where $1 = \{ () \}$ is the singleton set with the empty tuple () as sole inhabitant. Since powers are special cases of Cartesian products, they come with projection functions $\pi_i : X^n \to X$ and tuple functions $\langle f_1, \ldots, f_n \rangle : Y \to X^n$ for $n$ functions $f_i : Y \to X$. Finally, for a function $f : X \to Y$ we write $f^n : X^n \to Y^n$ for the obvious $n$-fold parallelisation of $f$.

**Exercises**

1.1.1 Check what a tuple function $\langle \pi_2, \pi_3, \pi_6 \rangle$ does on a product set $X_1 \times \cdots \times X_8$. What is the codomain of this function?

1.1.2 Check that, in general, the tuple function $\langle f_1, \ldots, f_n \rangle$ is the unique function $h : Y \to X_1 \times \cdots \times X_n$ with $\pi_i \circ h = f_i$ for each $i$.

1.1.3 Prove, using Equations (1.1) and (1.2) for tuples and projections, that:

$$(g_1 \times \cdots \times g_n) \circ (f_1 \times \cdots \times f_n) = (g_1 \circ f_1) \times \cdots \times (g_n \circ f_n).$$

1.1.4 Check that for each set $X$ there is a unique function $X \to 1$. Because of this property the set 1 is sometimes called ‘final’ or ‘terminal’. The unique function is often denoted by $!$.

Check also that a function $1 \to X$ corresponds to an element of $X$.

1.1.5 Define functions in both directions, using tuples and projections, that yield isomorphisms:

$$X \times Y \cong Y \times X \quad 1 \times X \cong X \quad X \times (Y \times Z) \cong (X \times Y) \times Z.$$ 

Try to use Equations (1.1) and (1.2) to prove these isomorphisms, without reasoning with elements.

1.2 Lists

The datatype of (finite) lists of elements from a given set is well-known in computer science, especially in functional programming. This section collects some basic constructions and properties.

For an arbitrary set $X$ we write $\mathcal{L}(X)$ for the set of all finite lists $[x_1, \ldots, x_n]$
of elements \( x_i \in X \), for arbitrary \( n \in \mathbb{N} \). Notice that we use square brackets \([\cdot]\) for lists, to distinguish them from tuples, which are typically written with round brackets \((\cdot,\cdot)\).

Thus, the set of lists over \( X \) can be defined as a union of all powers of \( X \).

\[
\mathcal{L}(X) := \bigcup_{n \in \mathbb{N}} X^n.
\]

When the elements of \( X \) are letters of an alphabet, then \( \mathcal{L}(X) \) is the set of words — the language — over this alphabet. The set \( \mathcal{L}(X) \) is alternatively written as \( X^* \), and called the Kleene star of \( X \).

We zoom in on some trivial cases. One has \( \mathcal{L}(0) = 1 \), since one can only form the empty word over the empty alphabet \( 0 = \emptyset \). If the alphabet contains only one letter, a word consists of a finite number of occurrences of this single letter. Thus:

\[
\mathcal{L}(1) \cong \mathbb{N}.
\]

We consider lists as an instance of what we call a collection data type, since \( \mathcal{L}(X) \) collects elements of \( X \) in a certain manner. What distinguishes lists from other collection types is that elements may occur multiple times, and that the order of occurrence matters. The lists \([a,b,a] \) and \([a,a,b] \) and \([a,b] \) differ. As we shall see later on, within a subset orders and multiplicities do not matter, see Section 1.3; and in a multiset the order of elements does not matter, but multiplicities do matter, see Section 1.4.

Let \( f : X \to Y \) be an arbitrary function. It can be used to map lists over \( X \) into lists over \( Y \) by applying \( f \) element-wise. This is what functional programmers call map-list. Here we like overloading, so we write \( \mathcal{L}(f) : \mathcal{L}(X) \to \mathcal{L}(Y) \) for this function, defined as:

\[
\mathcal{L}(f)([x_1,\ldots,x_n]) := [f(x_1),\ldots,f(x_n)].
\]

Thus, \( \mathcal{L} \) is an operation that not only sends sets to sets, but also functions to functions. It does so in such a way that identity maps and compositions are preserved:

\[
\mathcal{L}(\text{id}) = \text{id} \quad \mathcal{L}(g \circ f) = \mathcal{L}(g) \circ \mathcal{L}(f).
\]

We shall say: \( \mathcal{L} \) is functorial.

Functoriality can be used to define the marginal of a list on a product set, via \( \mathcal{L}(\pi_i) \), where \( \pi_i \) is a projection map. For instance, let \( \alpha \in \mathcal{L}(X \times Y) \) be of the form \( \alpha = [(x_1,y_1),\ldots,(x_n,y_n)] \). The first marginal \( \mathcal{L}(\pi_1)(\alpha) \in \mathcal{L}(X) \) is then computed as:

\[
\mathcal{L}(\pi_1)(\alpha) = \mathcal{L}(\pi_1)([(x_1,y_1),\ldots,(x_n,y_n)]) = [\pi_1(x_1,y_1),\ldots,\pi_1(x_n,y_n)] = [x_1,\ldots,x_n].
\]
1.2.1 Monoids

A monoid is a very basic mathematical structure, sometimes called semigroup. For convenience we define it explicitly.

**Definition 1.2.1.** A **monoid** consists of a set \( M \) with a special ‘unit’ element \( u \in M \) and a binary operation \( M \times M \to M \), written for instance as infix +, which is associative and has \( u \) has unit on both sides. That is, for all \( a, b, c \in M \),

\[
 a + (b + c) = (a + b) + c \quad \text{and} \quad u + a = a = a + u.
\]

The monoid is called **commutative** if \( a + b = b + a \), for all \( a, b \in M \). It is called **idempotent** if \( a + a = a \) for all \( a \in M \).

Let \( (M, u, +) \) and \( (N, v, \cdot) \) be two monoids. A function \( f : M \to N \) is called a **homomorphism of monoids** if \( f \) preserves the unit and binary operation, in the sense that:

\[
 f(u) = v \quad \text{and} \quad f(a + b) = f(a) \cdot f(b), \quad \text{for all} \ a, b \in M.
\]

For brevity we also say that such an \( f \) is a **map of monoids**, of simply a **monoid map**.

The natural numbers \( \mathbb{N} \) with addition form a commutative monoid \( (\mathbb{N}, 0, +) \). But also with multiplication they form a commutative monoid \( (\mathbb{N}, 1, \cdot) \). The function \( f(n) = 2^n \) is a homomorphism of monoids \( f : (\mathbb{N}, 0, +) \to (\mathbb{N}, 1, \cdot) \).

Various forms of collection types form monoids, with ‘union’ as binary operation. We start with lists, in the next result. The proof is left as (easy) exercise to the reader.

**Lemma 1.2.2.** 1. For each set \( X \), the set \( \mathcal{L}(X) \) of lists over \( X \) is a monoid, with the empty list \( [] \in \mathcal{L}(X) \) as unit element, and with concatenation \( ++ : \mathcal{L}(X) \times \mathcal{L}(X) \to \mathcal{L}(X) \) as binary operation:

\[
 [x_1, \ldots, x_n] ++ [y_1, \ldots, y_m] := [x_1, \ldots, x_n, y_1, \ldots, y_m].
\]

This monoid \( (\mathcal{L}(X), []) \) is neither commutative nor idempotent.

2. For each function \( f : X \to Y \) the associated map \( \mathcal{L}(f) : \mathcal{L}(X) \to \mathcal{L}(Y) \) is a homomorphism of monoids.

**1.2.2 Unit and flatten for lists**

Each element \( x \in X \) can be turned into a singleton list \([x] \in \mathcal{L}(X)\). We introduce a separate function \( \text{unit} : X \to \mathcal{L}(X) \) for this, so \( \text{unit}(x) := [x] \).

There is also a ‘flatten’ function which turns a list of lists into a list by
removing inner brackets $[\cdot]$. This function is written as $\text{flat} : \mathcal{L}(\mathcal{L}(X)) \to \mathcal{L}(X)$. It is defined as:

\[
\text{flat}([x_{11}, \ldots, x_{1m}], \ldots, [x_{k1}, \ldots, x_{km}]) := [x_{11}, \ldots, x_{1m}, \ldots, x_{k1}, \ldots, x_{km}].
\]

The next result contains some basic properties about unit and flatten. These properties will first be formulated in terms of equations, and then, alternatively as commuting diagrams. The latter style is preferred in this book.

**Lemma 1.2.3.** 1 For each function $f : X \to Y$ one has:

\[
\text{unit} \circ f = \mathcal{L}(f) \circ \text{unit} \quad \text{and} \quad \text{flat} \circ \mathcal{L}(L(f)) = \mathcal{L}(f) \circ \text{flat}.
\]

Equivalently, the following two diagrams commute.

\[
\begin{array}{ccc}
X & \overset{\text{unit}}{\longrightarrow} & \mathcal{L}(X) \\
\mathcal{L}(f) \downarrow & & \downarrow \mathcal{L}(f) \\
\mathcal{L}(\mathcal{L}(X)) & \overset{\text{flat}}{\longrightarrow} & \mathcal{L}(X)
\end{array}
\quad
\begin{array}{ccc}
\mathcal{L}(\mathcal{L}(X)) & \overset{\text{flat}}{\longrightarrow} & \mathcal{L}(X) \\
\mathcal{L}(\mathcal{L}(f)) \downarrow & & \downarrow \mathcal{L}(f) \\
\mathcal{L}(\mathcal{L}(Y)) & \overset{\text{flat}}{\longrightarrow} & \mathcal{L}(Y)
\end{array}
\]

2 One further has:

\[
\text{flat} \circ \text{unit} = \text{id} = \text{flat} \circ \mathcal{L}(\text{unit}) \quad \text{and} \quad \text{flat} \circ \text{flat} = \text{flat} \circ \mathcal{L}(\text{flat}).
\]

These two equations can equivalently be expressed via commutation of:

\[
\begin{array}{ccc}
\mathcal{L}(X) & \overset{\text{unit}}{\longrightarrow} & \mathcal{L}(\mathcal{L}(X)) \\
& \quad \downarrow \text{flat} & \quad \downarrow \mathcal{L}(\text{flat}) \\
\mathcal{L}(X) & \overset{\text{flat}}{\longrightarrow} & \mathcal{L}(\mathcal{L}(X)) \\
& \quad \downarrow \text{flat} & \\
& \quad \downarrow \mathcal{L}(\text{flat}) & \quad \downarrow \text{flat} \\
\mathcal{L}(\mathcal{L}(X)) & \overset{\text{flat}}{\longrightarrow} & \mathcal{L}(X)
\end{array}
\]

**Proof.** We shall do the first cases of each point, leaving the second cases to the interested reader. First, for $f : X \to Y$ and $x \in X$ one has:

\[
(\mathcal{L}(f) \circ \text{unit})(x) = \mathcal{L}(f)(\text{unit}(x)) = \mathcal{L}(f)([x]) = [f(x)] = \text{unit}(f(x)) = (\text{unit} \circ f)(x).
\]

Next, for the second point we take an arbitrary list $[x_1, \ldots, x_n] \in \mathcal{L}(X)$. Then:

\[
(\text{flat} \circ \text{unit})([x_1, \ldots, x_n]) = \text{flat}([[x_1, \ldots, x_n]]) = [x_1, \ldots, x_n]
\]

\[
(\mathcal{L}(\text{unit}) \circ \text{unit})([x_1, \ldots, x_n]) = \text{flat}([[\text{unit}(x_1), \ldots, \text{unit}(x_n)]) = \text{flat}([[x_1], \ldots, [x_n]]) = [x_1, \ldots, x_n].\]

\[\square\]
The equations in point 1 of this lemma are so-called naturality equations. They express that \( \text{unit} \) and \( \text{flat} \) work uniformly, independent of the set \( X \) involved. The equations in point 2 show that \( \mathcal{L} \) is a monad, see Section 1.10 for more information.

Exercises

1.2.1 Let \( X = \{a, b, c\} \) and \( Y = \{u, v\} \) be sets with a function \( f : X \to Y \) given by \( f(a) = u = f(c) \) and \( f(b) = v \). Write \( \alpha = [c, a, b, a] \) and \( \beta = [b, c, c, c] \). Compute consecutively:

- \( \alpha ++ \beta \)
- \( \beta ++ \alpha \)
- \( \alpha ++ (\beta ++ \alpha) \)
- \( (\alpha ++ \beta) ++ \alpha \)
- \( \mathcal{L}(f)(\alpha) \)
- \( \mathcal{L}(f)(\beta) \)
- \( \mathcal{L}(f)(\alpha) ++ \mathcal{L}(f)(\beta) \)
- \( \mathcal{L}(f)(\alpha ++ \beta) \).

1.2.2 We write \( \log \) for the logarithm function with base 2, so that \( \log(x) = y \) iff \( x = 2^y \). Verify that logarithm is a map of monoids:

\[
\begin{align*}
\log : (\mathbb{R}_{>0}, 1, \cdot) &\to (\mathbb{R}, 0, +) \\
\end{align*}
\]

Often the log function is used to simplify a computation, by turning multiplications into additions. Then one uses that \( \log \) is precisely this homomorphism of monoids. (An additional useful property is that \( \log \) is monotone: it preserves the order.)

1.2.3 Define a length function \( \text{len} : \mathcal{L}(X) \to \mathbb{N} \) on lists in the obvious way as \( \text{len}([x_1, \ldots, x_n]) = n \).

1. Prove that \( \text{len} \) is a map of monoids \( (\mathcal{L}(X), [], ++) \to (\mathbb{N}, 0, +) \).

2. Write \( ! \) for the unique function \( X \to 1 \) and check that \( \text{len} \) is \( \mathcal{L}(!) \).

Notice then that the previous point can then be seen as an instance of Lemma 1.2.2 (2).

1.2.4 Let \( X \) be an arbitrary set and \( (M, 0, +) \) an arbitrary monoid, with a function \( f : X \to M \). Prove that there is a unique homomorphism of monoids \( \overline{f} : (\mathcal{L}(X), [], ++) \to (M, 0, +) \) with \( \overline{f} \circ \text{unit} = f \).

This expresses that \( \mathcal{L}(X) \) is the free monoid on the set \( X \). The homomorphism \( \overline{f} \) is called the free extension of \( f \).
1.3 Powersets

The next collection type that will be studied is powerset $\mathcal{P}$. We will see that there are many similarities with lists $\mathcal{L}$ from the previous section.

For an arbitrary set $X$ we write $\mathcal{P}(X)$ for the set of all subsets of $X$ and $\mathcal{P}_{\text{fin}}(X)$ for the set of finite subsets. Thus:

$$\mathcal{P}(X) := \{U \mid U \subseteq X\} \quad \text{and} \quad \mathcal{P}_{\text{fin}}(X) := \{U \in \mathcal{P}(X) \mid U \text{ is finite}\}.$$  

If $X$ is a finite set itself, there is no difference between $\mathcal{P}(X)$ and $\mathcal{P}_{\text{fin}}(X)$. In the sequel we shall talk mostly about $\mathcal{P}$, but basically all properties of interest hold for $\mathcal{P}_{\text{fin}}$ as well.

First of all $\mathcal{P}$ is functorial. Given a function $f: X \to Y$ we can define a new function $\mathcal{P}(f): \mathcal{P}(X) \to \mathcal{P}(Y)$ by taking the image of $f$ on a subset. Explicitly, for $U \subseteq X$,

$$\mathcal{P}(f)(U) = \{f(x) \mid x \in U\}.$$  

The right-hand-side is clearly a subset of $Y$, and thus an element of $\mathcal{P}(Y)$. We have equations:

$$\mathcal{P}(\text{id}) = \text{id} \quad \mathcal{P}(g \circ f) = \mathcal{P}(g) \circ \mathcal{P}(f).$$  

Again we can use functoriality for marginalisation: for a subset (relation) $R \subseteq X \times Y$ on a product set we get its first marginal $\mathcal{P}(\pi_1)(R) \in \mathcal{P}(X)$ as subset:

$$\mathcal{P}(\pi_1)(R) = \{\pi_1(z) \mid z \in R\} = \{\pi_1(x, y) \mid (x, y) \in R\} = \{x \mid \exists y. (x, y) \in R\}.$$  

The next topic is the monoid structure on powersets.

**Lemma 1.3.1.** 1 For each set $X$, the powerset $\mathcal{P}(X)$ is a commutative and idempotent monoid, with empty subset $\emptyset \in \mathcal{P}(X)$ as neutral element and union $\cup$ of subsets of $X$ as binary operation.  

2 Each $\mathcal{P}(f): \mathcal{P}(X) \to \mathcal{P}(Y)$ is a map of monoids, for $f: X \to Y$.  

Next we define unit and flatten maps for powerset, much like for lists. The function $\text{unit}: X \to \mathcal{P}(X)$ sends an element to a singleton subset: $\text{unit}(x) := \{x\}$. The flatten function $\text{flat}: \mathcal{P}(\mathcal{P}(X)) \to \mathcal{P}(X)$ is given by union: for $A \subseteq \mathcal{P}(X)$,

$$\text{flat}(A) := \bigcup A = \{x \in X \mid \exists U \in A. x \in U\}.$$  

We mention, without proof, the following analogue of Lemma 1.2.3.
Lemma 1.3.2. 1 For each function $f : X \rightarrow Y$ the following ‘naturality’ diagrams commute.

$$
\begin{array}{ccc}
X & \xrightarrow{\text{unit}} & \mathcal{P}(X) \\
\downarrow f & & \downarrow \mathcal{P}(f) \\
Y & \xrightarrow{\text{unit}} & \mathcal{P}(Y)
\end{array}
\quad
\begin{array}{ccc}
\mathcal{P}(\mathcal{P}(X)) & \xrightarrow{\text{flat}} & \mathcal{P}(X) \\
\downarrow \mathcal{P}(f) & & \downarrow \mathcal{P}(f) \\
\mathcal{P}(\mathcal{P}(Y)) & \xrightarrow{\text{flat}} & \mathcal{P}(Y)
\end{array}
$$

2 Additionally, the ‘monad’ diagrams below commute.

$$
\begin{array}{ccc}
\mathcal{P}(X) & \xrightarrow{\text{unit}} & \mathcal{P}(\mathcal{P}(X)) \\
\downarrow \text{flat} & & \downarrow \text{flat} \\
\mathcal{P}(X) & & \mathcal{P}(\mathcal{P}(X))
\end{array}
\quad
\begin{array}{ccc}
\mathcal{P}(\mathcal{P}(\mathcal{P}(X))) & \xrightarrow{\text{flat}} & \mathcal{P}(\mathcal{P}(X)) \\
\downarrow \text{flat} & & \downarrow \text{flat} \\
\mathcal{P}(\mathcal{P}(X)) & \xrightarrow{\text{flat}} & \mathcal{P}(\mathcal{P}(X))
\end{array}
$$

1.3.1 From list to powerset

We have seen that lists and subsets behave in a similar manner. We strengthen this connection by defining a support function $\text{supp} : \mathcal{L}(X) \rightarrow \mathcal{P}_{\text{fin}}(X)$ between them, via:

$$
\text{supp}([x_1, \ldots, x_n]) := \{x_1, \ldots, x_n\}.
$$

Thus, the support of a list is the subset of elements occurring in the list. The support function removes order and multiplicities. The latter happens implicitly, via the set notation, above on the right-hand-side. For instance,

$$
\text{supp}([b, a, b, b, b]) = \{a, b\} = \{b, a\}.
$$

Notice that there is no way to go in the other direction, namely $\mathcal{P}_{\text{fin}}(X) \rightarrow \mathcal{L}(X)$. Of course, one can for each subset choose an order of the elements in order to turn the subset into a list. However, this process is completely arbitrary and is not uniform (natural).

The support function interacts nicely with the structures that we have seen so far. This is expressed in the result below, where we use the same notation $\text{unit}$ and $\text{flat}$ for different functions, namely for $\mathcal{L}$ and for $\mathcal{P}$: The context, and especially the types, of these operations will make clear which one is meant.

Lemma 1.3.3. Consider the support map $\text{supp} : \mathcal{L}(X) \rightarrow \mathcal{P}_{\text{fin}}(X)$ defined above.

1 It is a map of monoids ($\mathcal{L}(X), [], ++$) $\rightarrow$ ($\mathcal{P}(X), \emptyset, \cup$).
2 It is natural, in the sense that for \( f : X \to Y \) one has:

\[
\begin{array}{ccc}
\mathcal{L}(X) & \xrightarrow{\text{supp}} & \mathcal{P}_{\text{fin}}(X) \\
\downarrow \mathcal{L}(f) & & \downarrow \mathcal{P}_{\text{fin}}(f) \\
\mathcal{L}(Y) & \xrightarrow{\text{supp}} & \mathcal{P}_{\text{fin}}(Y)
\end{array}
\]

3 It commutes with the unit’s and flatten’s of list and powerset, as in:

\[
\begin{array}{ccc}
X & \xrightarrow{\text{unit}} & X \\
\downarrow \mathcal{L}(X) & \xleftarrow{\text{flat}} & \mathcal{L}(X) \\
\mathcal{P}_{\text{fin}}(X) & \xrightarrow{\text{supp}} & \mathcal{P}_{\text{fin}}(X)
\end{array}
\]

\[
\begin{array}{ccc}
X & \xrightarrow{\text{unit}} & X \\
\downarrow \mathcal{L}(X) & \xleftarrow{\text{flat}} & \mathcal{L}(X) \\
\mathcal{P}_{\text{fin}}(X) & \xrightarrow{\text{supp}} & \mathcal{P}_{\text{fin}}(X)
\end{array}
\]

Proof. The first point is easy and skipped. For point 2,

\[
\begin{align*}
(\mathcal{P}_{\text{fin}}(f) \circ \text{supp})([x_1, \ldots, x_n]) &= \mathcal{P}_{\text{fin}}(f)(\text{supp}([x_1, \ldots, x_n])) \\
&= \mathcal{P}_{\text{fin}}(f)([x_1, \ldots, x_n]) \\
&= \{f(x_1), \ldots, f(x_n)\} \\
&= \text{supp}([f(x_1), \ldots, f(x_n)]) \\
&= \text{supp}(\mathcal{L}(f)([x_1, \ldots, x_n])) \\
&= (\text{supp} \circ \mathcal{L}(f))(\{x_1, \ldots, x_n\}).
\end{align*}
\]

In point 3 commutation of the first diagram is easy:

\[
(\text{supp} \circ \text{unit})(x) = \text{supp}([x]) = \{x\} = \text{supp}(x).
\]

The second diagram requires a bit more work. Starting from a list of lists we get:

\[
\begin{align*}
(\text{flat} \circ \text{supp} \circ \mathcal{L}(\text{supp}))(\{[x_{11}, \ldots, x_{1, n_1}], \ldots, [x_{k1}, \ldots, x_{k, n_k}]\}) \\
= (\bigcup \circ \text{supp}((\text{supp}([x_{11}, \ldots, x_{1, n_1}]), \ldots, \text{supp}([x_{k1}, \ldots, x_{k, n_k}])))) \\
= (\bigcup \circ \text{supp}((\bigcup([x_{11}, \ldots, x_{1, n_1}], \ldots, [x_{k1}, \ldots, x_{k, n_k}])) \\
= \bigcup([x_{11}, \ldots, x_{1, n_1}], \ldots, [x_{k1}, \ldots, x_{k, n_k}]) \\
= \text{supp}([x_{11}, \ldots, x_{1, n_1}], \ldots, [x_{k1}, \ldots, x_{k, n_k}]) \\
= (\text{supp} \circ \text{flat})(([x_{11}, \ldots, x_{1, n_1}], \ldots, [x_{k1}, \ldots, x_{k, n_k}]).
\end{align*}
\]

1.3.2 Extraction

So far we have concentrated on how similar lists and subsets are: the only structural difference that we have seen up to now is that subsets form an idempotent.
and commutative monoid. But there are other important differences. Here we look at subsets of product sets, also known as relations.

The observation is that one can extract functions from a relation \( R \subseteq X \times Y \), namely functions of the form \( \text{extr}_1(R) : X \to \mathcal{P}(Y) \) and \( \text{extr}_2(R) : Y \to \mathcal{P}(X) \), namely:

\[
\text{extr}_1(R)(x) = \{ y \in Y \mid (x, y) \in R \}
\]

\[
\text{extr}_2(R)(y) = \{ x \in X \mid (x, y) \in R \}.
\]

In fact, one can easily reconstruct the relation \( R \) from \( \text{extr}_1(R) \), and also from \( \text{extr}_2(R) \), via:

\[
R = \{(x, y) \mid y \in \text{extr}_1(R)(x) \} = \{(x, y) \mid x \in \text{extr}_2(R)(y) \}.
\]

This all looks rather trivial, but such function extraction is less trivial for other data types, as we shall see later on, for distributions, where it will be called: disintegration.

In general, we write \( B^A \) for the set of functions from \( A \) to \( B \). Using this notation we can summarise the situation as: there are isomorphisms:

\[
\mathcal{P}(Y)^X \cong \mathcal{P}(X \times Y) \cong \mathcal{P}(X)^Y.
\] (1.3)

Functions of the form \( A \to \mathcal{P}(B) \) will later be called ‘channels’ from \( A \) to \( B \), see Section 1.7. What have just seen will then be described in terms of ‘extraction of channels’.

**Exercises**

1.3.1 Continuing Exercise 1.2.1, compute:

- \( \text{supp}(\alpha) \)
- \( \text{supp}(\beta) \)
- \( \text{supp}(\alpha + \beta) \)
- \( \text{supp}(\alpha) \cup \text{supp}(\beta) \)
- \( \text{supp}(\mathcal{L}(f)(\alpha)) \)
- \( \mathcal{P}_{\text{fin}}(f)(\text{supp}(\alpha)) \).

1.3.2 We have used finite unions \( (\emptyset, \cup) \) as monoid structure on \( \mathcal{P}(X) \) in Lemma 1.3.1(1). Intersections \( (X, \cap) \) give another monoid structure on \( \mathcal{P}(X) \).

1. Show that the negation/complement function \( \neg : \mathcal{P}(X) \to \mathcal{P}(X) \), given by:

\[
\neg U = X - U = \{ x \in X \mid x \not\in U \},
\]

is a homomorphism of monoids between \( (\mathcal{P}(X), \emptyset, \cup) \) and \( (\mathcal{P}, X, \cap) \).
1.4 Multisets

So far we have discussed two collection data types, namely lists and subsets of elements. In lists, elements occur in a particular order, and may occur multiple times (at different positions). Both properties are lost, when moving from lists to sets. In this section we look at multisets, which are ‘sets’ in which elements may occur multiple times. Hence multisets are somehow in between lists and subsets, since they do allow multiple occurrences, but no order.

The list and powerset examples are somewhat remote from probability theory. But multisets are much more directly relevant: first, because we use a similar notation for multisets and distribution; and second, because observed data can be organised nicely in terms of multisets. For instance, for statistical analysis, a document is often seen as a multiset of words, in which one keeps track of the words that occur in the document together with their frequency (multiplicity); in that case, the order of the words is ignored. Also, tables with observed data can be organised naturally as multisets; learning from such tables will be described in Section 1.5 as a (natural) transformation from multisets to distributions.

We start with an introduction about notation, terminology and conventions for multisets. Consider a set \( A = \{a, b, c\} \). An example of a multiset over \( A \) is:

\[
2|a\rangle + 5|b\rangle + 0|c\rangle
\]

In this multiset the element \( a \) occurs 2 times, \( b \) occurs 5 times, and \( c \) does not occur. The funny brackets \(|-\rangle\) are called ket notation; this is frequently used in quantum theory. Here it is meaningless notation, used to separate the natural numbers, used multiplicities, and the elements from \( A \).

Let \( X \) be an arbitrary set. Following the above ket-notation, a (finite) multiset

2 Prove that the intersections monoid structure is not preserved by maps \( \mathcal{P}(f): \mathcal{P}(X) \to \mathcal{P}(Y) \).

Hint: Look at preservation of the unit \( X \in \mathcal{P}(X) \).

1.3.3 Let \( X \) be an set and \( (M, 0, +) \) a commutative idempotent monoid, with a function \( f: X \to M \) between them. Prove that there is a unique homomorphism of monoids \( \overline{f}: (\mathcal{P}_{\text{fin}}(X), \emptyset, \cup) \to (M, 0, +) \) with \( \overline{f} \circ \text{unit} = f \).
over $X$ is an expression of the form:

$$n_1|x_1⟩ + \cdots + n_k|x_k⟩$$

where $n_i \in \mathbb{N}$ and $x_i \in X$.

This expression is a formal sum, not an actual sum (for instance in $\mathbb{R}$). We may write it as $\sum_i n_i|x_i⟩$. We use the convention:

- $0|x⟩$ may be omitted; but it may also be written explicitly in order to emphasise that the element $x$ does not occur in a multiset;
- a sum $n|x⟩ + m|x⟩$ is the same as $(n + m)|x⟩$;
- the order and brackets (if any) in a sum do not matter.

Thus, for instance, there is an equality of multisets:

$$2|a⟩ + (5|b⟩ + 0|c⟩) + 4|b⟩ = 9|b⟩ + 2|a⟩.$$

There is an alternative description of multisets. A multiset can be defined as a function $\varphi : X \rightarrow \mathbb{N}$ that has finite support. The support $\text{supp}(\varphi) \subseteq X$ is the set $\{x \in X \mid \varphi(x) \neq 0\}$. For each element $x \in X$ the number $\varphi(x) \in \mathbb{N}$ tells how often $x$ occurs in the multiset $\varphi$. Such a function $\varphi$ can also be written as a formal sum $\sum_x \varphi(x)|x⟩$, where $x$ ranges over $\text{supp}(\varphi)$.

For instance, the multiset $9|b⟩ + 2|a⟩$ over $A = \{a, b, c\}$ corresponds to the function $\varphi : A \rightarrow \mathbb{N}$ given by $\varphi(a) = 2$, $\varphi(b) = 9$, $\varphi(c) = 0$. Its support is thus $\{a, b\} \subseteq A$.

We shall freely switch back and forth between the ket-description and the function-description of multisets, and use whichever form is most convenient for the goal at hand.

Having said this, we stretch the idea of a multiset and do not only allow natural numbers $n \in \mathbb{N}$ as multiplicities, but also allow non-negative numbers $r \in \mathbb{R}_{\geq 0}$. Thus we can have a multiset of the form $\frac{1}{2}|a⟩ + \pi|b⟩$ where $\pi \in \mathbb{R}_{\geq 0}$ is the famous Archimedes’ constant. This added generality will be useful at times, although many examples of multisets will simply have natural numbers as multiplicities. We call such multisets natural.

**Definition 1.4.1.** For a set $X$ we shall write $M(X)$ for the set of all multisets over $X$. Thus, using the function approach:

$$M(X) \coloneqq \{\varphi : X \rightarrow \mathbb{R}_{\geq 0} \mid \text{supp}(\varphi) \text{ is finite}\}.$$  

The elements of $M(X)$ may be called mass functions, as in [89].

We shall write $N(X) \subseteq M(X)$ for the subset of ‘natural’ multisets, with natural numbers as multiplicities. Thus, $N(X)$ contains functions $\varphi \in M(X)$ with $\varphi(x) \in \mathbb{N}$, for all $x \in X$. 

---

14
1.4. Multisets

We shall write $\mathcal{M}(X)$ for the set of non-empty multisets. Thus:

$$
\mathcal{M}(X) := \{ \varphi \in \mathcal{M}(X) \mid \text{supp}(\varphi) \text{ is non-empty} \}
= \{ \varphi : X \to \mathbb{R}_{\geq 0} \mid \text{supp}(\varphi) \text{ is finite and non-empty} \}.
$$

Similarly, $\mathcal{N}(X) \subseteq \mathcal{M}(X)$ contains the non-empty natural multisets.

All of $\mathcal{M}, \mathcal{N}, \mathcal{M}^*, \mathcal{N}^*$ are functorial, in the same way. Hence we concentrate on $\mathcal{M}$. For a function $f : X \to Y$ we can define $M(f) : \mathcal{M}(X) \to \mathcal{M}(Y)$ in two equivalent ways:

$$
M(f)(\sum_{i} r_i | x_i) := \sum_{i} r_i f(x_i)
\quad \text{or as} \quad M(f)(\varphi)(y) := \sum_{x \in \text{supp}^{-1}(y)} \varphi(x).
$$

It may take a bit of effort to see that these descriptions are the same, see Exercise 1.4.1 below. Notice that in the sum $\sum_{i} r_i f(x_i)$ it may happen that $f(x_i) = f(x_j)$ for $x_i \neq x_j$, so that $r_i$ and $r_j$ are added. Thus, the support of $M(f)(\sum_{i} r_i | x_i)$ may have fewer elements than the support of $\sum_{i} r_i f(x_i)$, but the sum of all multiplicities is the same in $M(f)(\sum_{i} r_i | x_i)$ and $\sum_{i} r_i f(x_i)$.

**Lemma 1.4.2.** 1 The set $\mathcal{M}(X)$ of multisets over $X$ is a commutative monoid.

In functional form, addition and zero (unit) element $0 \in \mathcal{M}(X)$ are defined as:

$$(\varphi + \psi)(x) := \varphi(x) + \psi(x) \quad \text{and} \quad 0(x) := 0.
$$

These sums restrict to $\mathcal{N}(X)$.

2 The set $\mathcal{M}(X)$ is also a cone: it is closed under ‘scalar’ multiplication with non-negative numbers $r \in \mathbb{R}_{\geq 0}$, via:

$$(r \cdot \varphi)(x) := r \cdot \varphi(x).
$$

This scalar multiplication $r \cdot (-) : \mathcal{M}(X) \to \mathcal{M}(X)$ preserves the sums $(0, +)$ from the previous point, and is thus a map of monoids.

3 For each $f : X \to Y$, the function $M(f) : \mathcal{M}(X) \to \mathcal{M}(Y)$ is a map of monoids and also of cones. The latter means: $M(f)(r \cdot \varphi) = r \cdot M(f)(\varphi)$. \(\square\)

The element $0 \in \mathcal{M}(X)$ used in point 1 is the empty multiset. Similarly, the sum $+$ used there may be understood as an appropriate union of multisets. This union is implicit in the ket-notation. The set $\mathcal{N}(X)$ is not closed in general under scalar multiplication with $r \in \mathbb{R}_{\geq 0}$. It is closed under scalar multiplication with $n \in \mathbb{N}$, but such multiplications add nothing new since they can also be described via repeated addition.
Chapter 1. Collections and Channels

1.4.1 Tables of data as multisets

Consider the table (1.4) below where we have combined numeric information about blood pressure (either high $H$ or low $L$) and certain medicines (either type 1 or type 2 or no medicine, indicated as 0). There is data about 100 study participants:

<table>
<thead>
<tr>
<th>no medicine</th>
<th>medicine 1</th>
<th>medicine 2</th>
<th>totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>10</td>
<td>35</td>
<td>25</td>
</tr>
<tr>
<td>low</td>
<td>5</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>totals</td>
<td>15</td>
<td>45</td>
<td>40</td>
</tr>
</tbody>
</table>

We claim that we can capture this table as a (natural) multiset. To do so, we first form sets $B = \{H, T\}$ for blood pressure values, and $M = \{0, 1, 2\}$ for types of medicine. The above table can then be described as a multiset $\tau$ over the product $B \times M$, that is, as an element of $\tau \in N(B \times M)$, namely:

$$\tau = 10|H, 0\rangle + 35|H, 1\rangle + 25|H, 2\rangle + 5|L, 0\rangle + 10|L, 1\rangle + 15|L, 2\rangle.$$

We see that Table (1.4) contains 'totals' in its vertical and horizontal margins. They can be obtained from $\tau$ as marginals, using the functoriality of $N$. This works as follows.

$$N(\pi_1)(\tau) = 10|\pi_1(H, 0)\rangle + 35|\pi_1(H, 1)\rangle + 25|\pi_1(H, 2)\rangle$$
$$+ 5|\pi_1(L, 0)\rangle + 10|\pi_1(L, 1)\rangle + 15|\pi_1(L, 2)\rangle$$
$$= 10|H\rangle + 35|H\rangle + 25|H\rangle + 5|L\rangle + 10|L\rangle + 15|L\rangle$$
$$= 70|H\rangle + 30|L\rangle.$$

$$N(\pi_2)(\tau) = (10 + 5)|0\rangle + (35 + 10)|1\rangle + (25 + 15)|2\rangle$$
$$= 15|0\rangle + 45|1\rangle + 40|2\rangle.$$

In Section 1.5 we describe how to obtain probabilities from tables in a systematic manner.

1.4.2 Unit and flatten for multisets

As may be expected by now, there are also unit and flatten maps for multisets. The unit function $\text{unit}: X \rightarrow M(X)$ is simply $\text{unit}(x) := 1|x\rangle$. Flattening involves turning a multiset of multisets into a multiset. Concretely, this is done as:

$$\text{flat} \left( \frac{1}{2}|2|a\rangle + 2|c\rangle \right. + 5 \left| 1|b\rangle + \frac{1}{2}|e\rangle \right) = \frac{3}{2}|a\rangle + 5|b\rangle + \frac{5}{2}|c\rangle.$$
More generally, flattening is the function $\text{flat}: M(M(X)) \to M(X)$ with:

$$\text{flat} \left( \sum_i r_i \| \varphi_i \rangle \middle| x \right).$$

Notice that the big outer sum $\sum$ is a formal one, whereas the inner sum $\sum$ is an actual one, in $\mathbb{R}_{\geq 0}$, see the earlier example.

The following result does not come as a surprise anymore.

**Lemma 1.4.3.** 1 For each function $f: X \to Y$ one has:

$$X \xrightarrow{\text{unit}} M(X) \xrightarrow{M(f)} M(f) \xrightarrow{\text{flat}} M(X) \xrightarrow{\text{unit}} M(Y) \xrightarrow{M(f)} M(f) \xrightarrow{\text{flat}} M(Y)$$

2 Moreover:

$$M(X) \xrightarrow{\text{unit}} M(M(X)) \xrightarrow{M(\text{unit})} M(X) \xrightarrow{\text{flat}} M(X) \xrightarrow{\text{unit}} M(M(X)) \xrightarrow{M(\text{flat})} M(M(X)) \xrightarrow{\text{flat}} M(X) \xrightarrow{\text{flat}} M(X)$$

### 1.4.3 From list to multiset, and then to powerset

For a multiset $\varphi$ we have already used the ‘support’ definition $\text{supp}(\varphi) = \{ x \mid \varphi(x) \neq 0 \}$. This yields a map $\text{supp}: M(X) \to \mathcal{P}_{\text{fin}}(X)$, which is well-behaved, in the sense that it is natural and preserves the monoid structures on $M(X)$ and $\mathcal{P}_{\text{fin}}(X)$.

In the previous section we have seen a support map from lists $L$ to finite powerset $\mathcal{P}_{\text{fin}}$. This support map factorises through multisets, as described in the following triangle of different support functions.

$$L(X) \xrightarrow{\text{supp}} \mathcal{P}_{\text{list}}(X) \xrightarrow{\text{supp}} \mathcal{N}(X) \xrightarrow{\text{supp}} N(X)$$

What is missing is the support map $\text{supp}: L(X) \to N(X)$. Intuitively, it counts how many times an element occurs in a list, while ignoring the order of occurrences. Thus, for a list $\alpha \in L(X)$,

$$\text{supp}(\alpha)(x) := n \quad \text{if } x \text{ occurs } n \text{ times in the list } \alpha. \quad (1.6)$$

More concretely:

$$\text{supp}([a, b, a, b, c, b, b]) = 2|a\rangle + 4|b\rangle + 1|c\rangle.$$
The above diagram (1.3) expresses what we have stated informally at the beginning of this section, namely that multisets are somehow inbetween lists and subsets.

### 1.4.4 Max and argmax

Given a multiset we can determine the maximum multiplicity and the set of elements for which this maximum is reached. The latter is referred to as ‘argmax’. These notions are well-defined since we only consider finite multisets. Thus, for:

\[
\varphi = 4|a| + \frac{3}{2}|b| + 4|c| + 3|d|
\]

one has

\[
\begin{align*}
\max \varphi &= 4 \\
\argmax \varphi &= \{a, c\}.
\end{align*}
\]

This is formalised next.

**Definition 1.4.4.** For an arbitrary set \(X\) there are functions:

\[
M(X) \xrightarrow{\text{max}} \mathbb{R}_{\geq 0} \quad \text{and} \quad M(X) \xrightarrow{\argmax} \mathcal{P}_{\text{fin}}(X)
\]

defined as follows.

- \(\max \varphi\) is the least number \(s \in \mathbb{R}_{\geq 0}\) with \(\varphi(x) \leq s\), for all \(x \in X\); this means that \(\max \varphi\) is the supremum of the subset \(\{\varphi(x) \mid x \in X\} \subseteq \mathbb{R}_{\geq 0}\).
- \(\argmax \varphi = \{x \in X \mid \varphi(x) = \max \varphi\}\).

The following equivalent notations are used:

\[
\begin{align*}
\max \varphi &= \max_{x \in X} \varphi(x) \\
\argmax \varphi &= \argmax_{x \in X} \varphi(x).
\end{align*}
\]

For future use we collect some basic results about max and argmax. They are a bit technical and may be skipped at this stage until they are really needed.

**Lemma 1.4.5.** Let \(\varphi \in M(X), \psi \in M(Y)\) and \(\chi \in M(X, Y)\). For a binary operator \(\odot \in \{+, \cdot\}\) and for \(r \in \mathbb{R}_{\geq 0}\) we have the following results.

1. Maxima can be be taken separately in:

\[
\max_{x \in X, y \in Y} \varphi(x) \odot \psi(y) = \left( \max_{x \in X} \varphi(x) \right) \odot \left( \max_{y \in Y} \psi(y) \right)
\]

\[
\max_{x \in X} r \cdot \varphi(x) = r \cdot \left( \max_{x \in X} \varphi(x) \right)
\]

2. For argmax one has, when \(r > 0\),

\[
\begin{align*}
\argmax_{x \in X, y \in Y} \varphi(x) \odot \psi(y) &= \left( \argmax_{x \in X} \varphi(x) \right) \times \left( \argmax_{y \in Y} \psi(y) \right) \\
\argmax_{x \in X} r \cdot \varphi(x) &= \argmax_{x \in X} \varphi(x).
\end{align*}
\]
1.4. Multisets

3 For \((u, v) \in \arg\max \chi\) one has:

\[
\chi(u, v) = \max_{x \in X, y \in Y} \chi(x, y) = \max_{y \in Y} \chi(u, y) = \max_{x \in X} \chi(x, v).
\]

4 \[
\max_{x \in X, y \in Y} \varphi(x) \circ \chi(x, y) = \max_{y \in Y} \{ \varphi(x) \circ \chi(x, v) \mid x \in X, v \in \arg\max \chi(x, y) \}.
\]

5 \((u, v) \in \arg\max \varphi(x) \circ \chi(x, y) \iff u \in \arg\max \varphi(x) \circ \left( \max_{y \in Y} \chi(x, y) \right)
\]

\[
\text{and } v \in \arg\max \chi(u, y).
\]

**Proof.** The first two points are easy and left to the interested reader. For point (3) the only case that requires some care is:

\[
\max_{x \in X, y \in Y} \chi(x, y) \leq \max_{y \in Y} \chi(u, y).
\]

It is obtained as below, where the first equation holds by the fact that the pair

\((u, v)\) is in the argmax:

\[
\max_{x \in X, y \in Y} \chi(x, y) = \chi(u, v) \leq \max_{y \in Y} \chi(u, y).
\]

For point (4) we first abbreviate:

\[
r := \max_{x \in X, y \in Y} \varphi(x) \circ \chi(x, y)
\]

\[
s := \max_{y \in Y} \{ \varphi(x) \circ \chi(x, v) \mid x \in X, v \in \arg\max \chi(x, y) \}.
\]

We prove \(r = s\) via the next two inequalities.

\((r \leq s)\) For arbitrary \(x \in X, y \in Y\) we have:

\[
\chi(x, y) \leq \max_{y \in Y} \chi(x, z) = \chi(x, v)_{\ast} \quad \text{for each } v \in \arg\max_Y \chi(x, y).
\]

But then \(\varphi(x) \circ \chi(x, y) \leq \varphi(x) \circ \chi(x, v)_{\ast} \leq s\), and thus \(r \leq s\), since \(r\) is the least upperbound.

\((s \leq r)\) For \(x \in X\) and \(v_{\ast} \in \arg\max_Y \chi(x, y)\) we obviously have \(\varphi(x) \circ \chi(x, v_{\ast}) \leq r\) and thus \(s \leq r\).

Finally, for point (5) we need to prove two implications.

\((\Rightarrow)\) Let \((u, v) \in \arg\max_{x \in X, y \in Y} \varphi(x) \circ \chi(x, y)\). Then:

\[
\max_{x \in X} \varphi(x) \circ \max_{y \in Y} \chi(x, y) = \max_{x \in X} \varphi(x) \circ \chi(x, y)
\]

\[
= \varphi(u) \circ \chi(u, v)
\]

\[
= \max_{y \in Y} \varphi(u) \circ \chi(u, y) \quad \text{by point (3)}
\]

\[
= \varphi(u) \circ \max_{y \in Y} \chi(u, y)
\]
From this we may conclude:

\[ u \in \arg\max_{x \in X} \varphi(x) \odot \max_{y \in Y} \chi(x,y). \]

Using (3) once again yields:

\[ \varphi(u) \odot \chi(u,v) = \max_{y \in Y} \varphi(u) \odot \chi(u,y) = \varphi(u) \odot \max_{y \in Y} \chi(u,y) \]

By removing \( \varphi(u) \), via subtraction or division, we get \( \chi(u,v) = \max_{y \in Y} \varphi(u) \odot \chi(u,y) \), so that \( v \in \arg\max_{y \in Y} \chi(u,y) \).

(⇐) Let \( u \in \arg\max_{x \in X} \varphi(x) \odot \max_{y \in Y} \chi(x,y) \) and \( v \in \arg\max_{y \in Y} \chi(u,y) \).

Then \( \chi(u,v) = \max_{y \in Y} \chi(u,y) \) so that:

\[ \varphi(u) \odot \chi(u,v) = \varphi(u) \odot \max_{y \in Y} \chi(u,y) = \max_{x \in X} \varphi(x) \odot \max_{y \in Y} \chi(x,y) \quad \text{since } u \text{ is in the argmax} \]

Hence we are done.

\[ \square \]

1.4.5 Extraction

At the end of the previous section we have seen how to extract a function (channel) from a binary subset, that is, from a relation. It turns out that one can do the same for a binary multiset, that is, for a table. More specifically, there are isomorphisms:

\[ M(Y)^X \cong M(X \times Y) \cong M(X)^Y. \] (1.7)

This is analogous to (1.3) for powerset.

How does this work in detail? Suppose we have an arbitrary multiset/table \( \sigma \in M(X \times Y) \). From \( \sigma \) one can extract a function \( \text{extr}_1(\sigma) : X \to M(Y) \), and also \( \text{extr}_2(\sigma) : Y \to M(X) \), via:

\[ \text{extr}_1(\sigma)(x) = \sum_y \sigma(x,y)|y) \quad \text{extr}_2(\sigma)(y) = \sum_x \sigma(x,y)|x). \]

Notice that we are — conveniently — mixing ket and function notation for multisets. Conversely, \( \sigma \) can be reconstructed from \( \text{extr}_1(\sigma) \), and also from \( \text{extr}_2(\sigma) \), via \( \sigma(x,y) = \text{extr}_1(\sigma)(x)(y) = \text{extr}_2(\sigma)(y)(x) \).

Functions of the form \( A \to M(B) \) will also be used as channels \( A \to B \); see Section 1.7. That’s why we speak about ‘channel extraction’.
1.4. Multisets

As illustration, we apply extraction to the medicine - blood pressure Table 1.4 described as multiset $\tau \in M(B \times M)$. It gives rise to two channels $\text{extr}_1(\tau) : B \to M(M)$ and $\text{extr}_2(\tau) : M \to M(B)$. Explicitly:

\[ \text{extr}_1(\tau)(H) = \sum_{x \in M} \tau(H, x) |x\rangle = 10|0\rangle + 35|1\rangle + 25|2\rangle \]
\[ \text{extr}_1(\tau)(L) = \sum_{x \in M} \tau(L, x) |x\rangle = 5|0\rangle + 10|1\rangle + 15|2\rangle. \]

We see that this extracted function captures the two rows of Table 1.4. Similarly we get the columns via the second extracted function:

\[ \text{extr}_2(\tau)(0) = 10|L\rangle + 5|H\rangle \]
\[ \text{extr}_2(\tau)(1) = 35|L\rangle + 10|H\rangle \]
\[ \text{extr}_2(\tau)(2) = 25|L\rangle + 15|H\rangle. \]

Exercises

1.4.1 In the setting of Exercise 1.2.1, consider multisets $\varphi = 3|a\rangle + 2|b\rangle + 8|c\rangle$ and $\psi = 3|b\rangle + 1|c\rangle$. Compute:

- $\varphi + \psi$
- $\psi + \varphi$
- $M(f)(\varphi)$, both in ket-formulation and in function-formulation
- idem for $M(f)(\psi)$
- $M(f)(\varphi + \psi)$
- $M(f)(\varphi) + M(f)(\psi)$.

1.4.2 Consider, still in the context of Exercise 1.2.1, the ‘joint’ multiset $\varphi \in M(X \times Y)$ given by $\varphi = 2(a, u) + 3(a, v) + 5(c, v)$. Determine the marginals $M(\pi_1)(\varphi) \in M(X)$ and $M(\pi_2)(\varphi) \in M(Y)$.

1.4.3 Check that $M(\text{id}) = \text{id}$ and $M(g \circ f) = M(g) \circ M(f)$.

1.4.4 Prove Lemma 1.4.3.

1.4.5 Check that the support map $\text{supp} : L(X) \to M(X)$ is a homomorphism of monoids. Do the same for $\text{supp} : M(X) \to P_{\text{fin}}(X)$.

1.4.6 Prove that the support maps $\text{supp} : L(X) \to M(X)$ and $\text{supp} : M(X) \to P_{\text{fin}}(X)$ are natural: for an arbitrary function $f : X \to Y$ both rectangles below commute.

\[
\begin{array}{ccc}
L(X) & \xrightarrow{\text{supp}} & M(X) & \xrightarrow{\text{supp}} & P_{\text{fin}}(X) \\
L(f) & \downarrow \text{M(f)} & & \downarrow \text{P_{\text{fin}}(f)} & \\
L(Y) & \xrightarrow{\text{supp}} & M(Y) & \xrightarrow{\text{supp}} & P_{\text{fin}}(Y)
\end{array}
\]
Chapter 1. Collections and Channels

1.4.7 Show, along the lines of the proof of Lemma 1.4.5 (5) that:

\[(u, v, w) \in \text{argmax}_{x \in X, y \in Y, z \in Z} \varphi(x) \cdot \psi(x, y) \cdot \chi(y, z) \]

\[\iff u \in \text{argmax}_{x \in X} \left( \max_{y \in Y} \psi(x, y) \cdot \left( \max_{z \in Z} \chi(y, z) \right) \right) \quad \text{and} \quad v \in \text{argmax}_{y \in Y} \psi(u, y) \cdot \left( \max_{z \in Z} \chi(y, z) \right) \quad \text{and} \quad w \in \text{argmax}_{z \in Z} \chi(v, z). \]

1.4.8 Verify that the support map \(\text{supp}: M(X) \to \mathcal{P}_\text{fin}(X)\) commutes with extraction functions, in the sense that the following diagram commutes.

\[
\begin{array}{ccc}
M(X \times Y) & \xrightarrow{\text{supp}} & \mathcal{P}_\text{fin}(X \times Y) \\
\text{extr}_1 \downarrow & & \downarrow \text{extr}_1 \\
M(Y)^X & \xrightarrow{\text{supp}^*} & \mathcal{P}_\text{fin}(Y)^X
\end{array}
\]

Equationally, this amounts to showing that for \(\tau \in M(X \times Y)\) and \(x \in X\) one has:

\[\text{extr}_1(\text{supp}(\tau))(x) = \text{supp}(\text{extr}_1(\tau)(x)).\]

Here we use that \(\text{supp}^X(f) := \text{supp} \circ f\), so that \(\text{supp}^X(f)(x) = \text{supp}(f(x)).\)

1.4.9 Let \(X\) be a set and \((M, 0, +)\) a commutative monoid, with a function \(f: X \to M\).

1 Prove that there is a unique homomorphism \(\overline{f}: (N(X), 0, +) \to (M, 0, +)\) of monoids with \(\overline{f} \circ \text{unit} = f\).

2 Assume now that \(M\) is also a cone, i.e. comes with scalar multiplication \(r \cdot (-): M \to M\), for each \(r \in \mathbb{R}_{\geq 0}\), that is a homomorphism of monoids. Prove that there is now a unique homomorphism of cones \(\overline{f}: M(X) \to M\) with \(\overline{f} \circ \text{unit} = f\).

1.5 Probability distributions

This section is finally about probability, albeit in elementary form. It introduces finite discrete probability distributions, which we often simply call distributions or states. The notation and definitions that we use for distributions

\[\text{In the literature they are also called ‘multinomial’ or ‘categorical’ distributions, but we will not use that terminology.}\]
are very much like for multisets, except that multiplicities are required to add up to one.

This section follows to a large extend the structure of previous sections, in order to emphasise that distributions are not so different from lists, subsets and multisets. But there are also crucial differences: distributions do not form a monoid, but a ‘convex’ set, in which convex combinations can be formed.

A distribution, or a probability distribution, or a state, over a set $X$ is a finite formal convex sum of the form:

$$r_1 |x_1⟩ + \cdots + r_n |x_n⟩ \quad \text{where} \quad x_i \in X, r_i \in [0, 1] \text{ with } \sum r_i = 1.$$ 

We can write such an expression as a sum $\sum i r_i |x_i⟩$. It is called a convex sum since the $r_i$ add up to one. Thus, a distribution is a special ‘probabilistic’ multiset.

We write $D(X)$ for the set of distributions on a set $X$ of possible outcomes, so that $D(X) \subseteq M(X)$. Via this inclusion, we use the same conventions for distributions, as for multisets; they were described in the three bullet points in the beginning of Section 1.4. This set $X$ is often called the sample space, see e.g. [57].

For instance, for a coin we use the set $\{H, T\}$ with elements for head and tail as sample space. A fair coin is described on the left below, as a distribution over this set; the distribution on the right gives a coin with a slight bias.

$$\frac{1}{2}|H⟩ + \frac{1}{2}|T⟩ \quad 0.51|H⟩ + 0.49|T⟩.$$

In general, for a finite set $X = \{x_1, \ldots, x_n\}$ we use the Greek letter ‘upsilon’ for the uniform distribution $\upsilon_X \in D(X)$, given by $\upsilon_X = \sum_{i \in X} \frac{1}{n} |x_i⟩$. The above fair coin is a uniform distribution on the two-element set $\{H, T\}$. Similarly, a fair dice can be described as $\upsilon_{\text{pips}}$, where $\text{pips} = \{1, 2, 3, 4, 5, 6\}$. Figure 1.1 shows bar charts of several distributions.

**Example 1.5.1.** We shall explicitly described biased coins and binomials using the above formal convex sum notation.

1 The coin that we have seen above can be parametrised via a ‘bias’ probability $r \in [0, 1]$. The resulting coin is often called flip and is defined as:

$$\text{flip}(r) := r|1⟩ + (1-r)|0⟩$$

where 1 may understood as ‘head’ and 0 as ‘tail’. We may thus see flip as a function $\text{flip} : [0, 1] \to D([0, 1])$ from probabilities to distributions over the sample space $\{0, 1\}$ of Booleans.
Chapter 1. Collections and Channels

Figure 1.1 Plots of a slightly biased coin distribution $0.51|H⟩ + 0.49|T⟩$ and a fair (uniform) dice distribution on \{1, 2, 3, 4, 5, 6\} in the top row, together with a random distribution on the space \{0, 1, 2, \ldots, 24\} at the bottom.

For each number $K \in \mathbb{N}$ and probability $r \in [0, 1]$ there is the familiar binomial distribution $\text{binom}[K](r) \in \mathcal{D}(\{0, 1, \ldots, K\})$. It captures probabilities for iterated coin flips, and is given by the convex sum:

$$\text{binom}[K](r) := \sum_{0 \leq k \leq K} \binom{K}{k} \cdot r^k \cdot (1-r)^{K-k} |k⟩.$$

The multiplicity probability before $|k⟩$ in this expression is the chance of getting $k$ heads out of $K$ coin flips, where each flip has bias $r \in [0, 1]$. In this way we obtain a function $\text{binom}[K]: [0, 1] \to \mathcal{D}(\{0, 1, \ldots, K\})$. These multiplicities are plotted as bar charts in Figure 1.2 for two binomial distributions, both on the sample space \{0, 1, \ldots, 15\}.

We should check that $\text{binom}[K](r)$ really is a distribution, that is, that its multiplicities add up to one. This is a direct result of the binomial expansion theorem:

$$\sum_{0 \leq k \leq K} \binom{K}{k} \cdot r^k \cdot (1-r)^{K-k} = (r + (1-r))^K = 1^K = 1.$$

So far we have described distributions as formal convex sums. But they can
1.5. Probability distributions

be described, equivalently, in functional form. This is done in the definition below, which is much like Definition 1.4.1 for multisets. Also for distributions we shall freely switch between the above ket-formulation and the function-formulation given below.

Definition 1.5.2. The set \( \mathcal{D}(X) \) of all distributions over a set \( X \) can be defined as:

\[
\mathcal{D}(X) := \{ \omega: X \to [0, 1] \mid \text{supp}(\omega) \text{ is finite, and } \sum_x \omega(x) = 1 \}.
\]

Such a function \( \omega: X \to [0, 1] \), with finite support and values adding up to one, is often called a probability mass function, abbreviated as pmf.

This \( \mathcal{D} \) is functorial: for a function \( f: X \to Y \) we have \( \mathcal{D}(f): \mathcal{D}(X) \to \mathcal{D}(Y) \) defined either as:

\[
\mathcal{D}(f)(\sum_i r_i | x_i) := \sum_i r_i f(x_i)
\]

or as:

\[
\mathcal{D}(f)(\omega)(y) := \sum_{x \in f^{-1}(y)} \omega(x).
\]

A distribution of the form \( \mathcal{D}(f)(\omega) \in \mathcal{D}(Y) \), for \( \omega \in \mathcal{D}(X) \), is sometimes called an image distribution.

One has to check that the \( \mathcal{D}(f)(\omega) \) is a distribution again, that is, that its multiplicities add up to one. This works as follows.

\[
\sum_y \sum_{x \in f^{-1}(y)} \omega(x) = \sum_i \omega(x) = 1.
\]

We present two examples where functoriality of \( \mathcal{D} \) is frequently used, but not always in explicit form.

Example 1.5.3. 1 Computing marginals of ‘joint’ distributions involves functoriality of \( \mathcal{D} \). In general, one speaks of a joint distribution if its sample space is a product set, of the form \( X_1 \times X_2 \), or more generally, \( X_1 \times \cdots \times X_n \).
for \( n \geq 2 \). The \( i \)-th marginal of \( \omega \in \mathcal{D}(X_1 \times \cdots \times X_n) \) is defined as \( \mathcal{D}(\pi_i)(\omega) \), via the \( i \)-th projection function \( \pi_i: X_1 \times \cdots \times X_n \to X_i \).

For instance, the first marginal of the joint distribution, 
\[
\omega = \frac{1}{18}[H, 0] + \frac{1}{6}[H, 1] + \frac{1}{3}[H, 2] + \frac{1}{3}[T, 0] + \frac{1}{9}[T, 1] + \frac{1}{9}[T, 2]
\]
on the product space \([H, T] \times [0, 1, 2]\) is obtained as:
\[
\mathcal{D}(\pi_1)(\omega) = \frac{1}{18}[H] + \frac{1}{6}[H] + \frac{1}{3}[H] + \frac{1}{3}[T] + \frac{1}{9}[T] + \frac{1}{9}[T]
\]

Thus, \( \mathcal{D}(\omega) \) is defined as:
\[
\mathcal{D}(\pi_1)(\omega) = \frac{1}{3}[H] + \frac{1}{3}[T] \]

2 Let \( \omega \in \mathcal{D}(X) \) be distribution. In Chapter 2 we shall discuss random variables in detail, but here it suffices to know that it involves a function \( R: X \to \mathbb{R} \) from the sample space of the distribution \( \omega \) to the real numbers. Often, then, the notation 
\[
P[R = r] \in [0, 1]
\]
is used to indicate the probability that the random variable \( R \) takes value \( r \in \mathbb{R} \).

Since we now know that \( \mathcal{D} \) is functorial, we map apply it to the function \( R: X \to \mathbb{R} \). It gives another function \( \mathcal{D}(R): \mathcal{D}(X) \to \mathcal{D}(\mathbb{R}) \), so that \( \mathcal{D}(R)(\omega) \) is a distribution on \( \mathbb{R} \). We observe:
\[
P[R = r] = \mathcal{D}(R)(\omega)(r) = \sum_{x \in \mathbb{R} \cap (r)} \omega(x).
\]

Thus, \( P[R = (\cdot)] \) is an image distribution, on \( \mathbb{R} \). In the notation \( P[R = r] \) the distribution \( \omega \) on the sample space \( X \) is left implicit.

Here is concrete example. Recall that we write \( \text{pips} = \{1, 2, 3, 4, 5, 6\} \) for the sample space of a dice. Let \( M: \text{pips} \times \text{pips} \to \text{pips} \) take the maximum, so \( M(i, j) = \max(i, j) \). We consider \( M \) as a function \( \text{pips} \times \text{pips} \to \mathbb{R} \) via the inclusion \( \text{pips} \hookrightarrow \mathbb{R} \). Then, using the uniform distribution \( \nu \in \mathcal{D}(\text{pips} \times \text{pips}) \),
\[
P[M = k] = \text{the probability that the maximum of two dice throws is } k
\]
\[
= \mathcal{D}(M)(\nu)(k)
\]
\[
= \sum_{i, j \text{ with } \max(i, j) = k} \nu(i, j) = \sum_{i \leq k} \nu(i, k) + \sum_{j \leq k} \nu(k, j) = \frac{2k - 1}{36}.
\]

In the notation of this book we write the image distribution \( \mathcal{D}(M)(\nu) \) on \( \text{pips} \) as:
\[
\mathcal{D}(M)(\nu) = \frac{1}{36}[1] + \frac{1}{36}[2] + \frac{1}{36}[3] + \frac{1}{36}[4] + \frac{1}{36}[5] + \frac{11}{36}[6]
\]
\[
= P[M = 1][1] + P[M = 2][2] + P[M = 3][3]
\]
\[
\]
1.5. Probability distributions

Since in our notation the underlying (uniform) distribution \( \nu \) is explicit, we can also change it to another distribution \( \omega \in \mathcal{D}(\text{pips} \times \text{pips}) \) and still do the same computation. In fact, once we have seen product distributions in Section 1.8 we can compute \( \mathcal{D}(M)(\omega_1 \otimes \omega_2) \) where \( \omega_1 \) is a distribution for the first dice (say a uniform, fair one) and \( \omega_2 \) a distribution for the second dice (which may be unfair). This notation in which states are written explicitly gives much flexibility in what we wish to express and compute, see also Subsection 2.1.1 below.

At this stage one may expect a result saying that \( \mathcal{D}(X) \) forms a monoid, corresponding to the idea of taking sums of multisets. But that does not work. Instead, one can take convex sums of multisets. This works as follows. Suppose we have two distributions \( \omega, \rho \in \mathcal{D}(X) \) and a number \( s \in [0, 1] \). Then we can form a new distribution \( \sigma \in \mathcal{D}(X) \), as convex combination of \( \omega \) and \( \rho \), namely:

\[
\sigma := s \cdot \omega + (1 - s) \cdot \rho
\]

that is \( \sigma(x) = r \cdot \omega(x) + (1 - s) \cdot \rho(x) \).

At this stage we shall not axiomatise structures with such convex sums; they are sometimes called simply ‘convex sets’ or ‘barycentric algebras’, see [56] or [40] for details. A brief historical account occurs in [59, Remark 2.9].

1.5.1 Unit and flatten for distributions

The unit and flatten maps for multisets from Subsection 1.4.2 restrict to distributions. The unit function \( \text{unit} : X \to \mathcal{D}(X) \) is simply \( \text{unit}(x) := 1 \cdot |x\rangle \). Flattening is the function \( \text{flat} : \mathcal{D}(\mathcal{D}(X)) \to \mathcal{D}(X) \) with:

\[
\text{flat}\left( \sum_i r_i |\omega_i\rangle \right) := \sum_x \left( \sum_i r_i \cdot \omega_i(x) \right) |x\rangle.
\]

The only thing that needs to be checked is that the right-hand-side is again a convex sum, i.e. that its probabilities add up to one. This is easy:

\[
\sum_x \left( \sum_i r_i \cdot \omega_i(x) \right) = \sum_i r_i \cdot \sum_x \omega_i(x) = \sum_i r_i \cdot 1 = 1.
\]

The familiar properties of unit and flatten hold for distributions too: the analogue of Lemma 1.4.3 holds, with ‘multiset’ replaced by ‘distribution’.

We conclude with an observation about the size of supports of distributions.

Remark 1.5.4. We have restricted ourselves to finite probability distributions, by requiring that the support of a distribution is a finite set. This is fine in many situations of practical interest, such as in Bayesian networks. But there are relevant distributions that have infinite support, like the Poisson distribution.
**Chapter 1. Collections and Channels**

\(\text{pois}[\lambda]\) on \(\mathbb{N}\), with ‘mean’ or ‘rate’ parameter \(\lambda \geq 0\). It can be described as infinite formal convex sum:

\[
\text{pois}[\lambda] := \sum_{k \in \mathbb{N}} e^{-\lambda} \frac{\lambda^k}{k!} |k|.
\]  

(1.8)

This does not fit in \(\mathcal{D}(\mathbb{N})\). Therefore we sometimes use \(\mathcal{D}_{\infty}\) instead of \(\mathcal{D}\), where the finiteness of support requirement has been dropped:

\[
\mathcal{D}_{\infty}(X) := \{\varphi : X \to [0, 1] | \sum_x \varphi(x) = 1\}.
\]

This \(\mathcal{D}_{\infty}\) is also functorial and comes with unit and flat maps, just like for \(\mathcal{D}\).

Notice by the way that the multiplicities add up to one in the Poisson distribution because of the basic formula: \(e^\lambda = \sum_{k \in \mathbb{N}} \frac{\lambda^k}{k!}\). The Poisson distribution is typically used for counts of rare events.

**Exercises**

1.5.1 Check that a marginal of a uniform distribution is again a uniform distribution; more precisely, \(\mathcal{D}(\pi_1)(\nu_{X \times Y}) = \nu_X\).

1.5.2 1 Prove that \(\text{flip} : [0, 1] \to \mathcal{D}(2)\) is an isomorphism.

2 Check that \(\text{flip}(r)\) is the same as \(\text{binom}[1](r)\).

3 Describe the distribution \(\text{binom}[3](\frac{1}{4})\) concretely and interpret this distribution in terms of coin flips.

1.5.3 We often write \(n := \{0, \ldots, n-1\}\) so that 0 = \(\emptyset\), 1 = \(\{0\}\) and 2 = \([0, 1]\). Verify that:

\[
\mathcal{L}(0) \cong 1 \quad \mathcal{P}(0) \cong 1 \quad \mathcal{M}(0) \cong 1 \quad \mathcal{D}(0) \cong 0
\]

\[
\mathcal{L}(1) \cong \mathbb{N} \quad \mathcal{P}(1) \cong 2 \quad \mathcal{M}(1) \cong \mathbb{R}_{\geq 0} \quad \mathcal{D}(1) \cong 1
\]

\[
\mathcal{P}(n) \cong 2^n \quad \mathcal{M}(n) \cong (\mathbb{R}_{\geq 0})^n \quad \mathcal{D}(2) \cong [0, 1].
\]

The set \(\mathcal{D}(n+1)\) is often called the \(n\)-simplex. Describe it as a subset of \(\mathbb{R}^{n+1}\), and also as a subset of \(\mathbb{R}^n\).

1.5.4 Formulate and prove the analogue of Lemma 1.4.3 for \(\mathcal{D}\), instead of for \(\mathcal{M}\).

1.5.5 For a (non-zero) probability \(r \in (0, 1]\) one defines the geometric distribution \(\text{geom}[r] \in \mathcal{D}_{\infty}(\mathbb{N})\) as:

\[
\text{geom}[r] := \sum_{k \in \mathbb{N}_{\geq 0}} r \cdot (1-r)^{k-1} |k|.
\]

It captures the probability of being successful for the first time after \(k-1\) unsuccessful tries. Prove that this is a distribution indeed: its multiplicities add up to one.

**Hint:** Recall the summation formula for geometric series.
1.6. Frequentist learning: from multisets to distributions

1.5.6 Prove that a distribution \( \varphi \in \mathcal{D}_\infty(X) \) necessarily has countable support. 
*Hint:* Use that each set \( \{ x \in X \mid \varphi(x) > \frac{1}{n} \} \), for \( n > 0 \), can have only finitely many elements.

1.5.7 For \( N \in \mathbb{N} \), let’s write \( N[K](X) \subseteq \mathcal{M}(X) \) for the subset of natural multisets with \( K \in \mathbb{N} \) elements, that is, of \( \sum_i n_i x_i \) with \( \sum_i n_i = K \).

1. Check that \( N[K](2) \cong \{0, 1, \ldots, K\} \).

2. Re-describe the binomial distribution from Example 1.5.1(2) as a mapping

\[
\mathcal{D}(2) \xrightarrow{\text{binom}[K]} \mathcal{D}(N[K](2))
\]

3. Consider the ‘trinomial’ generalisation:

\[
\mathcal{D}(3) \xrightarrow{\text{trinom}[N]} \mathcal{D}(N[K](3))
\]

given by:

\[
\text{trinom}[K](r_00 \mid 0) + r_11 \mid 1 + r_22 \mid 2) = \sum_{k_0, k_1, k_2} K! \cdot k_0! \cdot k_1! \cdot k_2! \cdot r_0^{k_0} \cdot r_1^{k_1} \cdot r_2^{k_2} \cdot \binom{k_0}{0} \cdot \binom{k_1}{1} \cdot \binom{k_2}{2}
\]

Check that this yields a probability distribution. Interpret the trinomial distribution in analogy with the binomial distribution.

*Hint:* Use the multinomial theorem.

4. Generalise to ‘multinomial’ distributions:

\[
\mathcal{D}(m) \xrightarrow{\text{multinom}[K]} \mathcal{D}(N[K](m))
\]

(An answer will appear in Example 2.1.5.)

1.6 Frequentist learning: from multisets to distributions

In earlier sections we have seen several (natural) mappings between collection types, in the form of support maps, see the overview in Diagram (1.5). We are now introducing mappings \( \text{Flrn}: \mathcal{M}_e(X) \to \mathcal{D}(X) \) which are of a different nature. The name \( \text{Flrn} \) stands for ‘frequentist learning’, and may be pronounced as ‘eff-learn’. The frequentist interpretation of probability theory views probabilities as long term frequencies of occurrences. Here, these occurrences are given via multisets, which form the inputs of the \( \text{Flrn} \) function. Recall that \( \mathcal{M}_e(X) \) is the collection of non-empty multisets \( \sum_i r_i x_i \), with \( r_i \neq 0 \) for at
least one index \(i\). Equivalently one can require that the sum \(s := \sum_i r_i\) is non-zero.

The \(Flrn\) maps turns a (non-empty) multiset into a distribution, essentially by normalisation:

\[
Flrn(r_1|x_1) + \cdots + r_k|x_k) := \frac{r_1}{s}|x_1\rangle + \cdots + \frac{r_k}{s}|x_k\rangle
\]

where \(s := \sum_i r_i\). (1.9)

The normalisation step forces the sum on the right-hand-side to be a convex sum, with factors adding up to one. Clearly, from an empty multiset we cannot learn a distribution — technically because the above sum \(s\) is then zero so that we cannot divide by \(s\).

Using scalar multiplication from Lemma 1.4.2, we can define more succinctly:

\[
Flrn(\varphi) := \frac{1}{|\varphi|} \cdot \varphi \quad \text{where} \quad |\varphi| := \sum_i \varphi(x).
\]

We call this form \(Flrn\) of learning ‘frequentist’, following [48], because it boils down to counting.

**Example 1.6.1.** We present two illustrations of frequentist learning.

1. Suppose we have some coin of which the bias is unknown. There are experimental data showing that after tossing the coin 50 times, 20 times come up head (\(H\)) and 30 times yield tail (\(T\)). We can present these data as a *multiset* \(\sigma = 20|H\rangle + 30|T\rangle \in \mathcal{M}((H,T))\). When we wish to learn the resulting probabilities, we apply the frequentist learning map \(Flrn\) and get a *distribution* in \(\mathcal{D}((H,T))\), namely:

\[
Flrn(\sigma) = \frac{20}{20+30}|H\rangle + \frac{30}{20+30}|T\rangle = \frac{2}{5}|H\rangle + \frac{3}{5}|T\rangle.
\]

Thus, the bias (towards head) is \(\frac{2}{5}\). In this simple case we could have obtained this bias immediately from the data, but the \(Flrn\) map captures the general mechanism.

Notice that with frequentist learning, more (or less) consistent data gives the same outcome. For instance if we knew that 40 out of 100 tosses were head, or 2 out of 5, we would still get the same bias. Intuitively, these data give more (or less) confidence about the data. These aspects are not covered by frequentist learning, but by the more sophisticated form of ‘Bayesian’ learning. Another disadvantage of the rather primitive form of frequentist learning is that prior knowledge, if any, about the bias is not taken into account.
2 Recall the medical table (1.4) captured by the multiset \( \tau \in \mathcal{N}(B \times M) \). Learning from \( \tau \) yields the following joint distribution:

\[
F_{\text{Fln}}(\tau) = 0.1|H, 0) + 0.35|H, 1) + 0.25|H, 2) \\
+ 0.05|L, 0) + 0.1|L, 1) + 0.15|L, 2).
\]

Such a distribution, directly derived from a table, is sometimes called an empirical distribution [20].

In [32] it is argued that in general, people are not very good at probabilistic (esp. Bayesian) reasoning, but that they are much better at reasoning with “frequency formats”. To put it simply: the information that there is a 0.04 probability of getting a disease is more difficult to process than the information that 4 out of 100 people get the disease. In the current setting these frequency formats would correspond to natural multisets; they can be turned into distributions via the frequentist learning map \( F_{\text{Fln}} \). In the sequel we regularly return to the relationship between multisets and distributions in relation to various forms of learning, see esp. Chapter 4.

It turns out that the learning map \( F_{\text{Fln}} \) is ‘natural’, in a mathematical sense.

**Lemma 1.6.2.** The learning maps \( F_{\text{Fln}} : \mathcal{M}(X) \to \mathcal{D}(X) \) from (1.9) are natural in \( X \). This means that for each function \( f : X \to Y \) the following diagram commutes.

\[
\begin{array}{ccc}
\mathcal{M}(X) & \xrightarrow{F_{\text{Fln}}} & \mathcal{D}(X) \\
\downarrow{\mathcal{M}(f)} & & \downarrow{D(f)} \\
\mathcal{M}(Y) & \xrightarrow{F_{\text{Fln}}} & \mathcal{D}(Y)
\end{array}
\]

As a result, frequentist learning commutes with marginalisation.

**Proof.** Pick an arbitrary non-empty multiset \( \sigma = \sum_i r_i \langle x_i \rangle \) in \( \mathcal{M}(X) \) and write \( s := \sum_i r_i \). By non-emptiness of \( \sigma \) we have \( s \neq 0 \). Then:

\[
(F_{\text{Fln}} \circ \mathcal{M}(f))(\sigma) = F_{\text{Fln}}(\sum_i r_i \langle f(x_i) \rangle) \\
= \sum_i r_i \langle f(x_i) \rangle \\
= D(f)(\sum_i \frac{r_i}{s} \langle x_i \rangle) \\
= (D(f) \circ F_{\text{Fln}})(\sigma).
\]

We can apply this basic result to the medical data in Table (1.4), via multiset \( \tau \in \mathcal{N}(B \times M) \). We have already seen in Section 1.4 that the multiset-marginals \( \mathcal{N}(\pi_i)(\tau) \) produce the marginal columns and rows with totals. We can learn the distributions from the columns as:

\[
F_{\text{Fln}}(\mathcal{M}(\pi_1)(\tau)) = F_{\text{Fln}}(70|H) + |30)\langle L \rangle = 0.7|H) + 0.3|L).
\]
We can also take the distribution-marginal of the ‘learned’ distribution from the table, as described in Example 1.6.1 (2):

\[ M(\pi_1)(Flrn(\tau)) = (0.1 + 0.35 + 0.25)|H) + (0.05 + 0.1 + 0.15)|L) = 0.7|H) + 0.3|L). \]

Hence the basic operations of learning and marginalisations commute. This is a simple result, which many practitioners in probability are surely aware of, at an intuitive level, but maybe not in the mathematically precise form of Lemma 1.6.2.

An obvious next question is whether multiset-extraction — see the end of Section 1.4 — also commutes with ‘distribution-extraction’. The latter is called disintegration, and will be studied as a separate topic in Section 3.5; it will be related to frequentist learning in Section 4.1.

We conclude with an elementary operation in probability, which is commonly described as drawing an object from an urn. We shall describe it via frequentist learning Flrn. An urn typically contains finitely many objects, where the order does not matter, but multiplicities do matter. Hence we can describe the contents of an urn as a (natural) multiset \( \sigma \in \mathbb{N}(X) \), where \( X \) is the sample space of objects that may occur in the urn. For instance, we can use \( X = \{r, b\} \) when the urn contains red and blue balls.

For \( x \in X \) we define a draw operation \( D_x : \mathbb{N}(X) \rightarrow \mathbb{N}(X) \) which removes the element \( x \) from multiset \( \varphi \), if present. This draw is defined as:

\[
D_x(\varphi) = \begin{cases} 
\varphi & \text{if } \varphi(x) = 0, \text{ that is, if } x \text{ is not in } \varphi \\
(\varphi(x) - 1)|x) + \sum_{x' \neq x} \varphi(x')|x' \end{cases} \text{ otherwise.}
\]

These draws of individual elements are used in a general draw operation on a non-empty multiset. We describe its outcome as a distribution over pairs, consisting of the drawn element and the remaining multiset.

**Definition 1.6.3.** There is a draw operation:

\[
\mathbb{N}_*(X) \xrightarrow{D} \mathbb{D}(X \times \mathbb{N}(X))
\]

given by:

\[
D(\varphi) := \sum_{x \in \text{supp}(\varphi)} Flrn(\varphi(x)|x, D_x(\varphi)).
\]

For instance, suppose we have an urn/multiset \( \varphi = 2|r) + 1|b) \) with two red balls and 1 blue ball, then performing a draw yields a distribution:

\[
D(\varphi) = \frac{2}{3}|r, 1|r) + 1|b) \) + \frac{1}{3}|b, 2|r). \]
1.7. Channels

The first term describes the $\frac{2}{3}$ probability of drawing a red ball, with a multiset of one red and one blue ball remaining in the urn. The second term captures the $\frac{1}{3}$ probability of drawing a blue ball with two red ones remaining.

Exercises

1.6.1 Check that frequentist learning from a constant multiset yields a uniform distribution.

1.6.2 Show that Diagram [1.5] can be refined to:

$$
\begin{array}{c}
\mathcal{L}(X) \\
\text{supp} \\
\text{supp} \\
\mathcal{M}(X) \\
\text{Flrn} \\
\mathcal{D}(X) \\
\text{supp}
\end{array} \quad \xrightarrow{\text{supp}} \quad \mathcal{P}_{\text{fin}}(X) \quad \xrightarrow{\text{supp}} \quad \mathcal{M}^*(X)
$$

where $\mathcal{L}(X) \subseteq \mathcal{L}(X)$ is the subset of non-empty lists.

1.6.3 Show that frequentist learning commutes with units, i.e. that $\text{Flrn} \circ \text{unit} = \text{unit}$, but that it does not commute with flattens (like support commutes with flattens, in Lemma 1.3.3 (3)).

Hint: Start e.g. from $\Phi = \begin{bmatrix} 1 \mid 2\{a\} + 4\{c\} \} + 2 \mid 1\{a\} + 1\{b\} + 1\{c\} \} \end{bmatrix}$ in $\mathcal{M}(\mathcal{M}(\{a, b, c\}))$.

1.7 Channels

The previous sections covered the collection types of lists, subsets, multisets and distributions, with much emphasis on the similarities between them. In this section we will exploit these similarities in order to introduce the concept of channel, in a uniform approach, for all of these collection types at the same time. This will show that these data types are not only used for certain types of collections, but also for certain types of computation. Much of the rest of this book builds on the concept of a channel.

Let $T$ be be either $\mathcal{L}$, $\mathcal{P}$, $\mathcal{M}$ or $\mathcal{D}$. A state of type $Y$ is an element $\omega \in T(Y)$; it collects a number of elements of $Y$ in a certain way. In this section we abstract away from the particular type of collection. A channel, or sometimes $T$-channel, is a collection of states, parameterised by a set. Thus, a channel is a function of the form $c : X \rightarrow T(Y)$. Such a channel turns an element $x \in X$ into a certain collection $c(x)$ of elements of $Y$. Just like an ordinary function $f : X \rightarrow Y$ can be seen as a computation, we see a $T$-channel as a computation of type $T$. For instance, a channel $X \rightarrow \mathcal{P}(Y)$ is a non-deterministic computation, a channel $X \rightarrow \mathcal{M}(Y)$ is a resource-sensitive
computation, and a channel \( X \rightarrow \mathcal{D}(Y) \) is a probabilistic computation. We have already seen several examples of such probabilistic channels in Example 1.5.1, namely \( \text{flip} : [0, 1] \rightarrow \mathcal{D}([0, 1]) \) and \( \text{binom}[K] : [0, 1] \rightarrow \mathcal{D}([0, 1, \ldots, K]) \).

When it is clear from the context what \( T \) is, we often write a channel using functional notation, as \( c : X \rightarrow Y \), with a circle on the shaft of the arrow. Notice that a channel \( 1 \rightarrow X \) from the singleton set \( 1 = \{ 0 \} \) can be identified with a state on \( X \).

**Definition 1.7.1.** Let \( T \in \{ \mathcal{L}, \mathcal{P}, M, \mathcal{D} \} \), each of which is functorial, with its own flatten operation, as described in previous sections.

1 For a state \( \omega \in T(X) \) and a channel \( c : X \rightarrow T(Y) \) we can form a new state \( c \gg \omega \) of type \( Y \). It is defined as:

\[
c \gg \omega := (\text{flat} \circ T(c))(\omega) \quad \text{where} \quad T(X) \xrightarrow{T(c)} T(Y) \xrightarrow{\text{flat}} T(Y).
\]

This operation \( c \gg \omega \) is called **state transformation**, sometimes with addition: along the channel \( c \).

2 Let \( c : X \rightarrow Y \) and \( d : Y \rightarrow Z \) be two channels. Then we can compose them and get a new channel \( d \circ c : X \rightarrow Y \) via:

\[
(d \circ c)(x) := d \gg c(x) \quad \text{so that} \quad d \circ c = \text{flat} \circ T(d) \circ c
\]

We first look at some examples of state transformation.

**Example 1.7.2.** Take \( X = \{ a, b, c \} \) and \( Y = \{ u, v \} \).

1 For \( T = \mathcal{L} \) an example of a state \( \omega \in \mathcal{L}(X) \) is \( \omega = [c, b, h, a] \). An \( \mathcal{L} \)-channel \( f : X \rightarrow \mathcal{L}(Y) \) can for instance be given by:

\[
f(a) = [u, v] \quad f(b) = [u, u] \quad f(c) = [v, u, v].
\]

State transformation \( f \gg \omega \) using ‘map list’ via \( f \) and then flattening: turning a list of lists into a list:

\[
f \gg \omega = \text{flat}(\mathcal{L}(f)(\omega)) = \text{flat}([f(c), f(b), f(b), f(a)])
\]

\[
= \text{flat}([[v, u, v], [u, u], [u, u], [u, v]])
\]

\[
= [v, u, v, u, u, u, u, v, v].
\]

2 We consider the analogue example for \( T = \mathcal{P} \). We thus take as state \( \omega = [a, b, c] \) and as channel \( f : X \rightarrow \mathcal{P}(Y) \) with:

\[
f(a) = [u, v] \quad f(b) = [u] \quad f(c) = [u, v].
\]
3 For multisets a state in $M$ could be of the form $\omega = 3|a\rangle + 2|b\rangle + 5|c\rangle$ and a channel $f : X \to M(Y)$ could have:

$$f(a) = 10|u\rangle + 5|v\rangle \quad f(b) = 1|u\rangle \quad f(c) = 4|u\rangle + 1|v\rangle.$$  

We then get as state transformation:

$$f \gg \omega = \mathsf{flat}(M(f)(\omega))$$

$$= \mathsf{flat}(3f(a)) + 2f(b) + 5f(c))$$

$$= 30|u\rangle + 15|v\rangle + 2|u\rangle + 20|u\rangle + 5|v\rangle$$

$$= 52|u\rangle + 20|v\rangle.$$  

4 Finally, for distributions, let’s take as state $\omega = \frac{1}{6}|a\rangle + \frac{1}{2}|b\rangle + \frac{1}{3}|c\rangle \in D(X)$ and as $D$-channel $f : X \to D(Y)$,

$$f(a) = \frac{1}{2}|u\rangle + \frac{1}{2}|v\rangle \quad f(b) = 1|u\rangle \quad f(c) = \frac{3}{2}|u\rangle + \frac{1}{2}|v\rangle.$$  

We then get as state transformation:

$$f \gg \omega = \mathsf{flat}(D(f)(\omega))$$

$$= \mathsf{flat}(\frac{1}{6}f(a)) + \frac{1}{2}(f(b)) + \frac{1}{3}(f(c))$$

$$= \frac{1}{12}|u\rangle + \frac{1}{12}|v\rangle + \frac{1}{2}|u\rangle + \frac{1}{4}|u\rangle + \frac{1}{12}|v\rangle$$

$$= \frac{2}{3}|u\rangle + \frac{1}{3}|v\rangle.$$  

We now prove some general properties about channels.

**Lemma 1.7.3.** 1 Channel composition $\circ$ has a unit, namely $\text{unit} : Y \Rightarrow Y$, so that:

$$\text{unit} \circ c = c \quad \text{and} \quad d \circ \text{unit} = d,$$

for all channels $c : X \Rightarrow Y$ and $d : Y \Rightarrow Z$. Another way to write the second equation is: $d \gg \text{unit}(y) = d(y)$.

2 Channel composition $\circ$ is associative:

$$e \circ (d \circ c) = (e \circ d) \circ c$$

for all channels $c : X \Rightarrow Y$, $d : Y \Rightarrow Z$ and $e : Z \Rightarrow W$. 

3 State transformation via a composite channel is the same as two consecutive transformations:

\[(d \circ c) \triangleright \omega = d \triangleright (c \triangleright \omega).\]

4 Each ordinary function \(f : Y \to Z\) gives rise to a ‘trivial’ or ‘deterministic’ channel \(\langle f \rangle := \text{unit} \circ f : Y \to Z\). This construction \(\langle \cdot \rangle\) satisfies:

\[\langle f \rangle \triangleright \omega = T(f)(\omega),\]

where \(T\) is the type of channel involved. Moreover:

\[\langle g \rangle \circ \langle f \rangle = \langle g \circ f \rangle \quad \langle f \rangle \circ c = T(f) \circ c \quad d \circ \langle f \rangle = d \circ f.\]

for all functions \(g : Z \to W\) and channels \(c : X \to Y\) and \(d : Y \to W\).

**Proof.** We can give generic proofs, without knowing the type \(T \in \{\mathcal{L}, \mathcal{P}, \mathcal{M}, \mathcal{D}\}\) of channel, by using earlier results like Lemma 1.2.3, 1.3.2 and 1.4.3 and its analogue for \(\mathcal{D}\). Notice that we carefully distinguish channel composition \(\circ\) and ordinary function composition \(\circ\).

1 Both equations follow from the flat – unit law. By Definition 1.7.1(2):

\[\text{unit} \circ c = \text{flat} \circ T(\text{unit}) \circ c = \text{id} \circ c = c.\]

For the second equation we use naturality of unit in:

\[d \circ \text{unit} = \text{flat} \circ T(d) \circ \text{unit} = \text{flat} \circ \text{unit} \circ d = \text{id} \circ d = d.\]

2 The proof of associativity uses naturality and also the commutation of flatten with itself (the ‘flat – flat law’), expressed as \(\text{flat} \circ \text{flat} = \text{flat} \circ T(\text{flat})\).

\[e \circ (d \circ c) = \text{flat} \circ T(e) \circ (d \circ c) = \text{flat} \circ T(e) \circ \text{flat} \circ T(d) \circ c = \text{flat} \circ \text{flat} \circ T(T(e)) \circ T(d) \circ c = \text{flat} \circ T(T(\text{flat}) \circ T(T(e)) \circ T(d) \circ c = \text{flat} \circ T(\text{flat} \circ T(e) \circ d) \circ c = \text{flat} \circ T(e \circ d) \circ c = (e \circ d) \circ c\]
Along the same lines:

\[(d \circ c) \gg \omega = (\text{flat} \circ T(d \circ c))((\omega)\)
\[
= (\text{flat} \circ T(\text{flat} \circ T(d) \circ c))((\omega))
\[
= (\text{flat} \circ T(T(d)) \circ T(c))((\omega))
\]

by functoriality of T

by the flat – flat law

by naturality of flat

by naturality of unit

by a flat – unit law

by functoriality of T

by naturality of unit

by a flat – unit law

In the sequel we often omit writing the brackets \(\llbracket\cdots\rrbracket\) that turn an ordinary function \(f: X \to Y\) into a channel \(\langle f\rangle\). For instance, in a state transformation \(f \gg \omega\), it is clear that we use \(f\) as a channel, so that the expression should be read as \(\langle f\rangle \gg \omega\).

### 1.7.1 Graphical representation of probabilistic channels

Graphs, of various sorts, will play an important role in this book. A channel \(c: X \to Y\) can be represented as an edge from vertices \(X\) to \(Y\) in a directed graph. This representation will often be used. But there is a completely different representation of a probabilistic channel \(c: X \to Y\), namely as a transition graph. The elements \(x \in X\) and \(y \in Y\) will then be used as vertices. An edge
between them is a labeled arrow $x \xrightarrow{c(x)(y)} y$, where the number $c(x)(y) \in [0, 1]$ is seen as the probability of a transition from $x$ to $y$.

For instance, the state $\omega$ and channel $f: [a, b, c] \rightarrow [u, v]$ from Example 1.7.2 can be represented as a transition graph of the form:

\[
\begin{array}{c}
\text{a} \\
\downarrow \frac{1}{6} \\
\text{b} \\
\downarrow \frac{1}{2} \\
\text{u} \\
\text{c} \\
\downarrow \frac{1}{3} \\
\text{v} \\
\end{array}
\]

(1.11)

The black dot functions as initial node.

Exercises

1.7.1 Consider the $\mathcal{D}$-channel $f: [a, b, c] \rightarrow [u, v]$ from Example 1.7.2, together with a new $\mathcal{D}$-channel $g: [u, v] \rightarrow [1, 2, 3, 4]$ given by:

\[
g(u) = \frac{1}{4}|1\rangle + \frac{1}{4}|2\rangle + \frac{1}{4}|3\rangle + \frac{1}{4}|4\rangle \quad g(v) = \frac{1}{4}|1\rangle + \frac{1}{8}|3\rangle + \frac{1}{8}|3\rangle.
\]

Describe the general $g \circ f: [a, b, c] \rightarrow [1, 2, 3, 4]$ concretely.

1.7.2 Notice that a state of type $X$ can be identified with a channel $1 \rightarrow T(Y)$ with singleton set 1 as domain. Check that under this identification, state transformation $c \gg \omega$ corresponds to channel composition $c \circ \omega$.

1.7.3 Check that for $\mathcal{M}$- and $\mathcal{D}$-channels, state transformation can be described as:

\[
(c \gg \omega)(y) = \sum_x c(x)(y) \cdot \omega(x),
\]

that is, as:

\[
c \gg \omega = \sum_y \langle y | \left( \sum_x c(x)(y) \cdot \omega(x) \right) | y \rangle.
\]

where $c: X \rightarrow Y$ and $\omega$ a state of type $X$. We shall often use this formula.

2 Notice that we can also describe state transformation as a (convex) sum of scalar multiplications: that is, as:

\[
f \gg \omega = \sum_x \omega(x) \cdot c(x).
\]

38
1.7. Channels

1.7.4 Identify the channel \( f \) and the state \( \omega \) in Example 1.7.2 (4) with matrices:

\[
M_f = \begin{pmatrix}
\frac{1}{2} & 1 & \frac{1}{2} \\
\frac{1}{2} & 0 & \frac{1}{2}
\end{pmatrix}
M_\omega = \begin{pmatrix}
\frac{1}{2} \\
\frac{1}{2}
\end{pmatrix}
\]

Such matrices are sometimes called stochastic, since each of their columns has non-negative entries that add up to one.

1. Check that the matrix associated with the transformed state \( f \gg \omega \) is the matrix-column multiplication \( M_f \cdot M_\omega \).

(A general description appears in Remark 2.4.5.)

2. Check also what state transformation \( f \gg \omega \) means in the transition graph (1.11).

1.7.5 Draw transition graphs for the powerset and multiset channels in Example 1.7.2.

1.7.6 Prove that state transformation along \( D \)-channels preserves convex combinations, that is, for \( r \in [0, 1] \),

\[
f \gg (r \cdot \omega + (1 - r) \cdot \rho) = r \cdot (f \gg \omega) + (1 - r) \cdot (f \gg \rho).
\]

1.7.7 Let \( f : X \to Y \).

1. Prove that if \( f \) is a \( P_{fin} \)-channel, then the state transformation function \( f \gg (-) : P_{fin}(X) \to P_{fin}(Y) \) can also be defined via freeness, namely as the unique function \( f \) in Exercise 1.3.3.

2. Similarly, show that \( f \gg (-) = \bar{f} \) when \( f \) is a \( M \)-channel, as in Exercise 1.4.9.

1.7.8 Let \( c : X \to D(Y) \) be a \( D \)-channel and \( \sigma \in M(X) \) be a multiset. Because \( D(Y) \subseteq M(Y) \) we can also consider \( c \) as an \( M \)-channel, and use \( c \gg \sigma \). Prove that:

\[
Flrn(c \gg \sigma) = c \gg Flrn(\sigma) = Flrn(c) \gg Flrn(\sigma).
\]

1.7.9 1 Describe how (non-deterministic) powerset channels can be reversed, via a bijective correspondence between functions:

\[
\begin{align*}
X & \longrightarrow P(Y) \\
Y & \longrightarrow P(X)
\end{align*}
\]

(A description of this situation in terms of ‘dagger functors’ will appear in Example 3.8.1.)

2 Show that for finite sets \( X, Y \) there is a similar correspondence for multiset channels.
Chapter 1. Collections and Channels

3 Let \( X, Y \) be finite sets and \( c: X \to \mathcal{D}(Y) \) be a \( \mathcal{D} \)-channel. We can then define an \( \mathcal{M} \)-channel \( c^*: Y \to \mathcal{M}(X) \) by swapping arguments: \( c^*(y)(x) = c(x)(y) \). We call \( c \) a ‘bi-channel’ if \( c^* \) is also a \( \mathcal{D} \)-channel, i.e. if \( \sum_x c(x)(y) = 1 \) for each \( y \in Y \).

Prove that the identity channel is a bi-channel and that bi-channels are closed under composition.

4 Take \( A = \{a_0, a_1\} \) and \( B = \{b_0, b_1\} \) and define a channel \( \text{bell}: A \times 2 \to \mathcal{D}(B \times 2) \) as:

\[
\begin{align*}
\text{bell}(a_0, 0) &= \frac{1}{2}|b_0, 0\rangle + \frac{3}{8}|b_1, 0\rangle + \frac{1}{2}|b_1, 1\rangle \\
\text{bell}(a_0, 1) &= \frac{1}{2}|b_0, 1\rangle + \frac{1}{2}|b_1, 0\rangle + \frac{1}{2}|b_1, 1\rangle \\
\text{bell}(a_1, 0) &= \frac{3}{8}|b_0, 0\rangle + \frac{1}{2}|b_1, 0\rangle + \frac{3}{8}|b_1, 1\rangle \\
\text{bell}(a_1, 1) &= \frac{1}{2}|b_0, 0\rangle + \frac{3}{8}|b_1, 0\rangle + \frac{1}{2}|b_1, 1\rangle \\
\end{align*}
\]

Check that \( \text{bell} \) is a bi-channel.

(If captures the famous Bell table from quantum theory; we have deliberately used open spaces in the above description of the channel \( \text{bell} \) so that non-zero entries align, giving a ‘bi-stochastic’ matrix, from which one can read \( \text{bell}^* \) vertically.)

1.8 Parallel products

In the first section of this chapter we have seen Cartesian, parallel products \( X \times Y \) of sets \( X, Y \). Here we shall look at parallel products of states, and also at parallel products of channels. These new products will be written as tensors \( \otimes \). They express parallel combination. These tensors exist for \( \mathcal{P}, \mathcal{M} \) and \( \mathcal{D} \), but not for lists \( \mathcal{L} \). The reason for this absence will be explained below, in Remark 1.8.3. Increasingly, we are concentrating our attention on the probabilistic case.

**Definition 1.8.1.** Tensors, also called parallel products, will be defined first for states, for each type separately, and then for channels, in a uniform manner.

1 Let \( X, Y \) be arbitrary sets.

a For \( U \in \mathcal{P}(X) \) and \( V \in \mathcal{P}(Y) \) we define \( U \otimes V \in \mathcal{P}(X \times Y) \) as:

\[
U \otimes V := \{(x, y) \in X \times Y \mid x \in U \text{ and } y \in V\}.
\]

This product \( U \otimes V \) is often written simply as a product of sets \( U \times V \), but we prefer to have a separate notation for this product of subsets.
1.8. Parallel products

b For \( \varphi \in \mathcal{M}(X) \) and \( \psi \in \mathcal{M}(Y) \) we get \( \varphi \otimes \psi \in \mathcal{M}(X \times Y) \) as:

\[
\varphi \otimes \psi := \sum_{x,y} (\varphi(x) \cdot \psi(y)) |x,y\rangle
\]

that is \( (\varphi \otimes \psi)(x,y) = \varphi(x) \cdot \psi(y) \).

c For \( \omega \in \mathcal{D}(X) \) and \( \rho \in \mathcal{D}(Y) \) we get \( \omega \otimes \rho \in \mathcal{D}(X \times Y) \) as:

\[
\omega \otimes \rho := \sum_{x,y} (\omega(x) \cdot \rho(y)) |x,y\rangle
\]

that is \( (\omega \otimes \rho)(x,y) = \omega(x) \cdot \rho(y) \).

In this case it needs to be checked that the above expression forms a convex sum, but that’s easy.

2 Let \( c: X \to Y \) and \( d: A \to B \) be two channels, both of the same type \( T \in \{\mathcal{P}, \mathcal{M}, \mathcal{D}\} \). A channel \( c \otimes d: X \times A \to Y \times B \) is defined via:

\[
(c \otimes d)(x,a) := c(x) \otimes d(a).
\]

The right-hand-side uses the tensor product of the appropriate type \( T \), as defined in the previous point.

We shall use tensors not only in binary form \( \otimes \), but also in \( n \)-ary form \( \otimes \cdots \otimes \), both for states and for channels.

We see that tensors \( \otimes \) involve the products \( \times \) of the underlying domains. A simple illustration of a (probabilistic) tensor product is:

\[
\left(\frac{1}{3}|u\rangle + \frac{1}{3}|v\rangle + \frac{1}{3}|w\rangle\right) \otimes \left(\frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle\right) = \frac{1}{6}|u,0\rangle + \frac{1}{6}|u,1\rangle + \frac{1}{6}|v,0\rangle + \frac{1}{6}|v,1\rangle + \frac{1}{6}|w,0\rangle + \frac{1}{6}|w,1\rangle.
\]

These tensor products tend to grow really quickly in size, since the number of entries of the two parts have to be multiplied.

Parallel products are well-behaved, as described below.

**Lemma 1.8.2.** 1 Transformation of parallel states via parallel channels is the parallel product of the individual transformations:

\[
(c \otimes d) \gg (\omega \otimes \rho) = (c \gg \omega) \otimes (d \gg \rho).
\]

2 Parallel products of channels interact nicely with unit and composition:

\[
\text{unit} \otimes \text{unit} = \text{unit} \quad (c_1 \otimes d_1) \circ (c_2 \otimes d_2) = (c_1 \circ c_2) \otimes (d_1 \circ d_2).
\]

3 The tensor of trivial, deterministic channels is obtained from their product:

\[
\langle f \rangle \otimes \langle g \rangle = \langle f \times g \rangle
\]

where \( f \times g = \langle f \circ \pi_1, g \circ \pi_2 \rangle \), see Subsection 1.1.1.
Proof. We shall do the proofs for \( M \) and for \( D \) of point (1) at once, using Exercise 1.7.3 (1), and leave the remaining cases to the reader.

\[
((c \otimes d) \gg (\omega \otimes \rho))(y, b) = \sum_{x,a} (c \otimes d)(x, a)(y, b) \cdot (\omega \otimes \rho)(x, a)
\]

\[
= \sum_{x,a} (c(x) \otimes d(a))(y, b) \cdot (\omega(x) \cdot \rho(a))
\]

\[
= (\sum_x c(x)(y) \cdot \omega(x)) \cdot (\sum_a d(a)(b) \cdot \rho(a))
\]

\[
= (c \gg \omega)(y) \cdot (d \gg \rho)(b)
\]

\[
= (c \gg \omega) \otimes (d \gg \rho)(y, b).
\]

As promised we look into why parallel products don’t work for lists.

Remark 1.8.3. Suppose we have two list \([a, b, c]\) and \([u, v]\) and we wish to form their parallel product. Then there are many ways to do so. For instance, two obvious choices are:

\[
\begin{align*}
[(a, u), (a, v), (b, u), (b, v), (c, u), (c, v)] \\
[(a, u), (b, u), (c, u), (a, v), (b, v), (c, v)]
\end{align*}
\]

There are many other possibilities. The problem is that there is no canonical choice. Since the order of elements in a list matter, there is no commutativity property which makes all options equivalent, like for multisets. Technically, the tensor for \( L \) does not exist because \( L \) is not a commutative (i.e. monoidal) monad; this is an early result in category theory going back to [60].

We illustrate the use of parallel composition for iterated drawing.

Example 1.8.4. Recall the draw operation \( D : \mathcal{N}(X) \to \mathcal{D}(X \times \mathcal{N}(X)) \) from Definition 1.6.3. We will ignore the non-emptyness requirement and consider this operation as a probabilistic channel:

\[
\mathcal{N}(X) \xrightarrow{D} X \times \mathcal{N}(X)
\]

If we wish to draw twice, consecutively, we can describe this as composition of channels, namely as a probabilistic channel:

\[
\mathcal{N}(X) \xrightarrow{D} X \times \mathcal{N}(X) \xrightarrow{\text{id} \otimes D} X \times X \times \mathcal{N}(X)
\]

Concretely, for a multiset \( \varphi \in \mathcal{N}(X) \) with at least two elements:

\[
((\text{id} \otimes D) \circ D)(\varphi) = \sum_{x, y \in X} \text{Flrn}(\varphi)(x) \cdot \text{Flrn}(D_x(\varphi)(y)) \cdot 1 \cdot x, y, D_y(D_x(\varphi))
\]
For instance, for an urn with two red \((r)\) and one blue \((b)\) ball:

\[
\begin{align*}
\left((\text{id} \otimes D) \circ D\right)(2|r\rangle + 1|b\rangle) & = (\text{id} \otimes D) \gg (\frac{1}{2}|r, 1|b\rangle + \frac{1}{2}|b, 1|b\rangle + \frac{1}{2}|b, 1|r\rangle) \\
& = \frac{1}{2}|r, 1|b\rangle + \frac{1}{2}|r, b, 1|r\rangle + \frac{1}{2}|b, 1|b\rangle + \frac{1}{2}|b, 1|r\rangle.
\end{align*}
\]

Next we illustrate the use of parallel products \(\otimes\) of probabilistic channels for a standard ‘summation’ property of Poisson distributions \(\text{pois}\), see (1.8). This property is commonly expressed in terms of random variables \(X\) as: if \(X_1 \sim \text{pois}[\lambda_1]\) and \(X_2 \sim \text{pois}[\lambda_2]\) then \(X_1 + X_2 \sim \text{pois}[\lambda_1 + \lambda_2]\). We have not discussed random variables yet, but we do not need them for the channel-based reformulation below.

Recall that the Poisson distribution has infinite support, so that we need to use \(\mathcal{D}_\infty\) instead of \(\mathcal{D}\), see Remark 1.5.4, but that difference is immaterial here. Tensors work for \(\mathcal{D}_\infty\) too. We now use the mapping \(\lambda \mapsto \text{pois}[\lambda]\) as a function \(\mathbb{R}_{\geq 0} \to \mathcal{D}_\infty(\mathbb{N})\) and as a channel \(\text{pois}: \mathbb{R}_{\geq 0} \to \mathbb{N}\).

**Lemma 1.8.5.** Poisson commutes with finite sums, in the sense that the following diagrams of channels commute,

\[
\begin{array}{ccc}
\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} & \xrightarrow{\text{pois} \circ \text{pois}} & \mathbb{R}_{\geq 0} \\
\downarrow & & \downarrow \\
\mathbb{N} \times \mathbb{N} & \xrightarrow{\text{pois}} & \mathbb{N}
\end{array}
\]

\[
\begin{array}{ccc}
1 & \xrightarrow{0} & \mathbb{R}_{\geq 0} \\
\downarrow & & \downarrow \\
1 & \xrightarrow{0} & \mathbb{N}
\end{array}
\]

in which the sums \((+, 0)\) are used (twice) as a deterministic channels. Notice, by the way, that we have put the circle \(\circ\) that is typical for channels on the arrows in these rectangles.

This can be read as: the Poisson channel is a homomorphism of commutative monoids, from \((\mathbb{R}_{\geq 0}, +, 0)\) to \((\mathbb{N}, +, 0)\).

**Proof.** We first do the above diagram on the left, for which we pick arbitrary \(\lambda_1, \lambda_2 \in \mathbb{R}_{\geq 0}\) and \(k \in \mathbb{N}\). We start with the south-east path in the rectangle and
will arrive at the east-south path:

\[
( + \circ (\text{pois} \otimes \text{pois}))(\lambda_1, \lambda_2)(k) = D(+(\text{pois}[\lambda_1] \otimes \text{pois}[\lambda_2])(k))
\]
\[
= \sum_{k_1, k_2, \lambda_1 + \lambda_2 = k} (\text{pois}[\lambda_1] \otimes \text{pois}[\lambda_2])(k_1, k_2)
\]
\[
= \sum_{m \leq k} \text{pois}[\lambda_1](m) \cdot \text{pois}[\lambda_2](k - m)
\]
\[
= \sum_{m \leq k} \left( e^{-\lambda_1} \cdot \frac{\lambda_1^m}{m!} \right) \left( e^{-\lambda_2} \cdot \frac{\lambda_2^{k-m}}{(k-m)!} \right)
\]
\[
= \frac{e^{-\lambda_1 - \lambda_2}}{k!} \cdot \sum_{m \leq k} \left( \frac{k}{m} \right) \cdot \lambda_1^m \cdot \lambda_2^{k-m}
\]
\[
= \frac{e^{-\lambda_1 - \lambda_2}}{k!} \cdot (\lambda_1 + \lambda_2)^k
\]
\[
= \text{pois}[\lambda_1 + \lambda_2](k)
\]
\[
= (\text{pois} \circ +)(\lambda_1, \lambda_2)(k).
\]

Commutation of the above diagram on the right, describing preservation of zero, amounts to the trivial equation \(\text{pois}[0] = 1[0]\).

The fact that the Poisson channel forms a homomorphism of monoids requires the wider perspective of category theory, which allows us to say that both \((\mathbb{R}_{\geq 0}, +, 0)\) and \((\mathbb{N}, +, 0)\) are monoids in the symmetric monoidal category of \(D_{\infty}\)-channels. The interested reader is referred to the literature for details, see e.g. [4, 68], since this is beyond the current goals.

1.8.1 Marginalisation and entwinedness

In Section 1.7 we have seen projection maps \(\pi_i : X_1 \times \cdots \times X_n \to X_i\) for Cartesian products of sets. Since these \(\pi_i\) are ordinary functions, they can be turned into channels \(\langle \pi_i \rangle = \text{unit} \circ \pi_i : X_1 \times \cdots \times X_n \to X_i\). We will be sloppy and typically omit these brackets \(\langle \cdot \rangle\) for projections. The kind of arrow, \(\to\) of \(\rightarrow\), or the type of operation at hand, will then indicate whether \(\pi_i\) is used as a function or as channel.

The next result only works in the probabilistic case, for \(D\). The exercises below will provide counterexamples for \(P\) and \(M\).

**Lemma 1.8.6.** 1 Projection channels take parallel products of probabilistic
1.8. Parallel products

states apart, that is, for \( \omega_i \in \mathcal{D}(X_i) \) we have:

\[
\pi_i \gg (\omega_1 \otimes \cdots \otimes \omega_n) = \mathcal{D}(\pi_i)(\omega_1 \otimes \cdots \otimes \omega_n) = \omega_i.
\]

Thus, marginalisation of parallel product yields its components.

2 Similarly, projection channels commute with parallel products of probabilistic channels, in the following manner:

\[
\begin{array}{c}
X_1 \times \cdots \times X_n \\
\pi_i \downarrow \\
X_i \\
\end{array}
\begin{array}{c}
\otimes \mathcal{C} \\
\pi_i \\
\end{array}
\begin{array}{c}
\rightarrow \\
Y_1 \times \cdots \times Y_n \\
\end{array}
\begin{array}{c}
\pi_i \downarrow \\
Y_i \\
\end{array}
\]

Proof. We only do point (1), since point (2) then follows easily, using that parallel products of channels are defined pointwise, see Definition 1.8.1 (2). The first equation in point (1) follows from Lemma 1.7.3 (4), which yields \( \pi_i \gg (-) = \mathcal{D}(\pi_i)(-) \). We restrict to the special case where \( n = 2 \) and \( i = 1 \). Then:

\[
\begin{align*}
\mathcal{D}(\pi_1)(\omega_1 \otimes \omega_2)(x) &= \sum_y (\omega_1 \otimes \omega_2)(x, y) & \text{see Definition 1.5.2} \\
&= \sum_y \omega_1(x) \cdot \omega_2(y) \\
&= \omega_1(x) \cdot (\sum_y \omega_2(y)) = \omega_1(x) \cdot 1 = \omega_1(x). \\
\end{align*}
\]

The last line of this proof relies on the fact that probabilistic states (distributions) involve a convex sum, with multiplicities adding up to one. This does not work for multisets, see Exercise 1.8.5 below.

We introduce special, post-fix notation for marginalisation via ‘masks’.

Definition 1.8.7. A mask \( M \) is a finite list of 0’s and 1’s, that is an element \( M \in \mathcal{L}(\{0, 1\}) \). For a state \( \omega \) of type \( T \in \{\mathcal{L}, \mathcal{P}, M, \mathcal{D}\} \) on \( X_1 \times \cdots \times X_n \) and a mask \( M \) of length \( n \) we write:

\[
\omega M
\]

for the marginal with mask \( M \). Informally, it keeps all the parts from \( \omega \) at a position where there is 1 in \( M \) and it projects away parts where there is 0. This is best illustrated via an example:

\[
\omega[1, 0, 1, 0, 1] = T((\pi_1, \pi_3, \pi_5))(\omega) \in T(X_1 \times X_3 \times X_5)
\]

\[
= (\pi_1, \pi_3, \pi_5) \gg \omega.
\]

More generally, for a channel \( c : Y \rightarrow X_1 \times \cdots \times X_n \) and a mask \( M \) of length \( n \) we use pointwise marginalisation via the same postcomposition:

\[
cM \text{ is the channel } y \mapsto c(y)M.
\]
With Cartesian products one can take an arbitrary tuple $t \in X \times Y$ apart into $\pi_1(t) \in X, \pi_2(t) \in Y$. By re-assembling these parts the original tuple is recovered: $\langle \pi_1(t), \pi_2(t) \rangle = t$. This does not work for collections: a joint state is typically not the parallel product of its marginals, see Example 1.8.9 below.

We introduce a special name for this.

**Definition 1.8.8.** A joint state $\omega$ on $X \times Y$ is called *non-entwined* if it is the parallel product of its marginals:

$$\omega = \omega[1,0] \otimes \omega[0,1].$$

Otherwise it is called *entwined*. This notion of entwinedness may be formulated with respect to $n$-ary states, via a mask, see for example Exercise 1.8.1, but it may then need some re-arrangement of components.

Lemma 1.8.6(1) shows that a probabilistic product state of the form $\omega_1 \otimes \omega_2$ is non-entwined. But in general, joint states are entwined, so that the different parts are correlated and can influence each other. This is a mechanism that will play an important role in the sequel.

**Example 1.8.9.** Take $X = \{u,v\}$ and $A = \{a,b\}$ and consider the state $\omega \in \mathcal{D}(X \times A)$ given by:

$$\omega = \frac{1}{8}|u,a\rangle + \frac{3}{8}|u,b\rangle + \frac{3}{8}|v,a\rangle + \frac{1}{8}|v,b\rangle.$$ 

We claim that $\omega$ is entwined. Indeed, $\omega$ has first and second marginals $\omega[1,0] = \mathcal{D}(\pi_1)(\omega) \in \mathcal{D}(X)$ and $\omega[0,1] = \mathcal{D}(\pi_2)(\omega) \in \mathcal{D}(A)$, namely:

$$\omega[1,0] = \frac{3}{8}|u\rangle + \frac{3}{8}|v\rangle \quad \text{and} \quad \omega[0,1] = \frac{1}{2}|a\rangle + \frac{1}{2}|b\rangle.$$ 

The original state $\omega$ differs from the product of its marginals:

$$\omega[1,0] \otimes \omega[0,1] = \frac{3}{16}|u,a\rangle + \frac{3}{16}|u,b\rangle + \frac{5}{16}|v,a\rangle + \frac{5}{16}|v,b\rangle.$$ 

This entwined follows from a general characterisation, see Exercise 1.8.7 below.

### 1.8.2 Copying

For an arbitrary set $X$ there is a *copy* function $\Delta_n: X \rightarrow X^n = X \times \cdots \times X$ ($n$ times), given as:

$$\Delta_n(x) := [x, \ldots, x] \quad (n \text{ times } x).$$

We often omit the subscript $n$, when it is clear from the context, especially when $n = 2$. This copy function can be turned into a copy channel $\langle \Delta_n \rangle =$
1.8. Parallel products

For two channels $c: X \to X^n$, but, recall, we often omit writing $\preceq \to$ for simplicity. These $\Delta_n$ are alternatively called copiers or diagonals.

As functions one has $\pi_i \circ \Delta_n = \text{id}$, and thus as channels $\pi_i \circ \Delta_n = \text{unit}$. State transformation with a copier $\Delta \gg \omega$ is different for taking a tensor product $\omega \otimes \omega$, as the following simple example illustrates. For clarity we now write the brackets $\preceq \to$ explicitly. First,

$$\langle \Delta \gg \langle \frac{1}{2}(0) + \frac{3}{2}(1) \rangle \rangle = \sum_{x,y} \langle \Delta \gg \langle \frac{1}{2}(0) + \frac{3}{2}(1) \rangle \rangle \langle x,y \rangle$$

$$= \sum_{x,y} \langle \text{unit}(\Delta(0))(x,y) \cdot \frac{1}{2} + \text{unit}(\Delta(1))(x,y) \cdot \frac{3}{2} \rangle \langle x,y \rangle$$

$$= \sum_{x,y} \langle \text{unit}(0,0)(x,y) \cdot \frac{1}{2} + \text{unit}(1,1)(x,y) \cdot \frac{3}{2} \rangle \langle x,y \rangle$$

$$= \frac{1}{2}(0,0) + \frac{3}{2}(1,1).$$

In contrast:

$$\langle \frac{1}{2}(0) + \frac{3}{2}(1) \rangle \otimes \langle \frac{1}{2}(0) + \frac{3}{2}(1) \rangle = \frac{1}{2}(0,0) + \frac{3}{2}(0,1) + \frac{3}{2}(1,0) + \frac{3}{2}(1,1).$$

One might expect a commuting diagram like in Lemma 1.8.6 for copiers, but that does not work: channel do not commute with diagonals, see Exercise 1.8.8 below for a counterexample. Copiers do commute with deterministic channels of the form $\langle f \rangle = \text{unit} \circ f$, as in:

$$\Delta \circ \langle f \rangle = \langle (f \otimes f) \circ \Delta \rangle \circ \Delta \quad \text{because} \quad \Delta \circ f = (f \times f) \circ \Delta.$$  

In fact, this commutation property may be used as definition for a channel to be deterministic, see Exercise 1.8.11.

For an ordinary function $f: X \to Y$ one can form the graph of $f$ as the relation $gr(f) \subseteq X \times Y$ given by $gr(f) = \{(x,y) \mid f(x) = y\}$. There is a similar thing that one can do for channels, given a state. But please keep in mind that this kind of ‘graph’ has nothing to do with the graphical models that we will be using throughout, like Bayesian networks or string diagrams.

**Definition 1.8.10.** 1 For two channels $c: X \to Y$ and $d: X \to Z$ with the same domain we can define a tuple channel:

$$\langle c, d \rangle := (c \otimes d) \circ \Delta : X \to Y \times Z$$

2 In particular, for a state $\sigma$ on $Y$ and a channel $d: Y \to Z$ we define the graph as the joint state on $Y \times Z$ defined by:

$$gr(\sigma, d) := \langle \text{id}, d \rangle \gg \sigma \quad \text{so that} \quad gr(\sigma, d)(y, z) = \sigma(y) \cdot d(y)(z).$$
Please note that we are overloading the tuple notation \(<c, d>\). Above we use it for the tuple of channels, so that \(<c, d>\) has type \(X \rightarrow \mathcal{D}(Y \times Z)\). Interpreting \(<c, d>\) as a tuple of functions, like in Subsection 1.1.1 would give a type \(X \rightarrow \mathcal{D}(Y) \times \mathcal{D}(Z)\). For channels we standardly use the channel-interpretation for the tuple.

In Subsection 1.4.4 we have looked at maxima for multiplicities of multisets, via the functions \(\text{max} \) and \(\text{argmax}\). These functions can be applied to distributions as well, via the inclusions \(\mathcal{D}(X) \subseteq \mathcal{M}(X)\). Often we are interested in finding such maxima in joint states, especially when they are of the graph form \(\text{gr}(\sigma, d)\) as described above.

**Lemma 1.8.11.** For a channel \(d: Y \rightarrow Z\) and a state \(\sigma \in \mathcal{D}(Y)\) one has:

1. \(\text{max} \left(\text{gr}(\sigma, d)\right) = \text{max}\{\sigma(y) \cdot d(y)(z) \mid y \in Y, z \in \text{argmax} d(y)\}\)
2. \((u, v) \in \text{argmax} \left(\text{gr}(\sigma, d)\right) \iff u \in \text{argmax} \tau(y) \cdot \max_{y \in X} d(y)
\equiv u \in \text{argmax} \sigma(y) \cdot \max_{y \in X} d(u)\) and \(v \in \text{argmax} d(u)\).

**Proof.** Directly by Lemma 1.4.5 (4) and (5). \(\square\)

The formulation in the above second point forms the basis for a mini-algorithm for computing \(\text{argmax} \left(\text{gr}(\sigma, d)\right)\), namely:

1. compute for each \(y \in Y\) the maximum \(m_y \equiv \max d(y) \in \mathbb{R}_{\geq 0}\);
2. determine the subset \(\text{argmax}_y \sigma(y) \cdot m_y \subseteq Y\);
3. for each \(u\) in this subset, determine the subset \(\text{argmax}_u d(u) \subseteq Z\);
4. return such pairs \((u, v) \in Y \times Z\) where \(u \in \text{argmax}_y \sigma(y) \cdot m_y\) and \(v \in \text{argmax}_u d(u)\).

The important thing is that this can be done without actually using the (possibly big) joint state \(\text{gr}(\sigma, d)\). It applies in particular in Bayesian networks.

**Exercises**

1.8.1 Check that the distribution \(\omega \in \mathcal{D}([a, b] \times [a, b] \times [a, b])\) given by:

\[
\omega = \frac{1}{25}[aaa] + \frac{1}{12}[aab] + \frac{1}{12}[aba] + \frac{1}{6}[abb] + \frac{1}{6}[bba] + \frac{1}{12}[bab] + \frac{1}{12}[bbb]
\]

satisfies:

\[
\omega = \omega[1, 1, 0] \otimes \omega[0, 0, 1].
\]

1.8.2 Check that for finite sets \(X, Y\), the uniform distribution on \(X \times Y\) is non-entwined — see also Exercise 1.5.1.
1.8. Parallel products

1.8.3 Describe, in the style of Example 1.8.4, how to draw three balls from the urn/multiset $2|\omega = 1|b)$. 

1.8.4 Check that:

$$\omega[0, 1, 1, 0, 1, 1][0, 1, 1, 0] = \omega[0, 0, 1, 0, 1, 0]$$

What is the general result behind this?

1.8.5 Let $U \in \mathcal{P}(X)$ and $V \in \mathcal{P}(Y)$. Show that the equation

$$(U \otimes V)[1, 0] = U$$

does not hold in general, in particular not when $V = \emptyset$ (and $U \neq \emptyset$).

1.8.6 Similarly, check that for arbitrary $\varphi \in \mathcal{M}(X)$ and $\psi \in \mathcal{M}(Y)$ one does not have:

$$(\varphi \otimes \psi)[0, 1] = \varphi.$$  

It fails when $\psi$ is but $\varphi$ isn’t the empty multiset $\emptyset$. But it also fails in non-zero cases, e.g. for the multisets $\varphi = 2|a| + 4|b| + 1|c|$ and $\psi = 3|a| + 2|v|$. Check this by doing the required computations.

1.8.7 Let $X = \{u, v\}$ and $A = \{a, b\}$ as in Example 1.8.9 Prove that a state $\omega = r_1|u, a| + r_2|u, b| + r_3|v, a| + r_4|v, b| \in \mathcal{D}(X \times A)$, where $r_1 + r_2 + r_3 + r_4 = 1$, is non-entwined if and only if $r_1 \cdot r_4 = r_2 \cdot r_3.$

1.8.8 Consider the probabilistic channel $f : X \rightarrow Y$ from Example 1.7.2 (4) and show that on the other hand $\Delta \circ f : X \rightarrow Y \times Y$ is given by:

- $a \mapsto \frac{1}{2}|u, u| + \frac{1}{2}|v, v|$  
- $b \mapsto |u, u| + \frac{1}{2}|v, v|$  
- $c \mapsto \frac{3}{4}|u, u| + \frac{1}{4}|v, v|.$

On the other hand, $(f \otimes f) \circ \Delta : X \rightarrow Y \times Y$ is described by:

- $a \mapsto \frac{1}{2}|u, u| + \frac{1}{2}|u, v| + \frac{1}{2}|v, u| + \frac{1}{2}|v, v|$  
- $b \mapsto |u, u| + \frac{1}{2}|v, v|$  
- $c \mapsto \frac{3}{16}|u, u| + \frac{3}{16}|u, v| + \frac{3}{16}|v, u| + \frac{1}{16}|v, v|.$

1.8.9 Check that for ordinary functions $f : X \rightarrow Y$ and $g : X \rightarrow Z$ one can relating channel tupling and ordinary tupling via the operation $\langle \cdot, \cdot \rangle$ as:

$$\langle \langle f, g \rangle = \langle f, g \rangle = \text{unit} \circ \langle f, g \rangle \text{ i.e. } \langle \text{unit} \circ f, \text{unit} \circ g \rangle = \text{unit} \circ \langle f, g \rangle.$$  

Conclude that the copy channel $\Delta$ is $\langle \text{unit}, \text{unit} \rangle$. 

49
1.8.10 Check that the tuple of channels from Definition 1.8.10 satisfies both:
\[ \pi_i \otimes (c_1, c_2) = c_i \quad \text{and} \quad (\pi_1, \pi_2) = \text{unit} \]
but not, essentially as in Exercise 1.8.8,
\[ \langle c_1, c_2 \rangle \circ d = \langle c_1 \circ d, c_2 \circ d \rangle. \]

1.8.11 Let \( c : X \to Y \) commute with diagonals, in the sense that \( \Delta \circ \cdot = (c \otimes c) \circ \Delta \). Prove that \( c \) is deterministic, i.e., of the form \( c = \text{unit} \circ f \) for a function \( f : X \to Y \).

1.8.12 Consider the joint state \( \text{Flrn}(\tau) \) from Example 1.6.1 (2).

1.8.13 Prove that for multisets \( \sigma \in M(X) \) and \( \tau \in M(Y) \) one has:
\[ \text{Flrn}(\sigma \otimes \tau) = \text{Flrn}(\sigma) \otimes \text{Flrn}(\tau) \]

1.8.14 Consider the bi-channel \( \text{bell} : A \times 2 \to B \times 2 \) from Exercise 1.7.9 (4).

1.9 A Bayesian network example

This section uses the previous sections, especially their probabilistic parts, to give semantics for a Bayesian network example. At this stage we just describe...
1.9. A Bayesian network example

an example, without giving the general approach, but we hope that it clarifies what is going on and illustrates to the reader the relevance of (probabilistic) channels and their operations. A general description of what a Bayesian network is appears much later, in Definition 3.1.2.

The example that we use is a standard one, copied from the literature, namely from [20]. Bayesian networks were introduced in [80], see also [5, 8, 9, 57, 61].

Consider the diagram/network in Figure 1.3. It is meant to capture probabilistic dependencies between several wetness phenomena in the oval boxes. For instance, in winter it is more likely to rain (than when it’s not winter), and also in winter it is less likely that a sprinkler is on. Still the grass may be wet by a combination of these occurrences. Whether a road is slippery depends on rain, not on sprinklers.

The letters \(A, B, C, D, E\) in this diagram are written exactly as in [20]. Here they are not used as sets of Booleans, with inhabitants \(true\) and \(false\), but instead we use these sets with elements:

\[
A = \{a, a^\perp\} \quad B = \{b, b^\perp\} \quad C = \{c, c^\perp\} \quad D = \{d, d^\perp\} \quad E = \{e, e^\perp\}
\]

The notation \(a^\perp\) is read as ‘not \(a\)’. In this way the name of an elements suggests to which set the element belongs.

The diagram in Figure 1.3 becomes a Bayesian network when we provide it with conditional probability tables. For the lower three nodes they look as follows:

<table>
<thead>
<tr>
<th>(\text{winter} )</th>
<th>(a)</th>
<th>(a^\perp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1/5)</td>
<td>(2/5)</td>
<td></td>
</tr>
</tbody>
</table>

| \(\text{sprinkler} \) | \(A\) | \(b\) | \(b^\perp\) |
|---------------------|-----|-----|
| \(a\) | \(1/5\) | \(4/5\) |
| \(a^\perp\) | \(3/4\) | \(1/4\) |

| \(\text{rain} \) | \(A\) | \(c\) | \(c^\perp\) |
|-------------------|-----|-----|
| \(a\) | \(4/5\) | \(1/5\) |
| \(a^\perp\) | \(1/10\) | \(9/10\) |
And for the upper two nodes we have:

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>C</th>
<th>d</th>
<th>d'</th>
</tr>
</thead>
<tbody>
<tr>
<td>wet grass</td>
<td>b c</td>
<td>9/20</td>
<td>1/20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b' c</td>
<td>9/10</td>
<td>1/10</td>
<td></td>
</tr>
<tr>
<td>slippery road</td>
<td>c e</td>
<td>7/10</td>
<td>3/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c'</td>
<td>3/5</td>
<td>1/5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b c</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

This Bayesian network is thus given by nodes, each with a conditional probability table, describing likelihoods in terms of previous ‘ancestor’ nodes in the network (if any).

How to interpret all this data? How to make it mathematically precise? It is not hard to see that the first ‘winter’ table describes a probability distribution on the set $A$, which, in the notation of this book, is given by:

$$wi = \frac{3}{5}a + \frac{2}{5}a' .$$

Thus we are assuming with probability of 60% that we are in a winter situation. This is often called the prior distribution, or also the initial state.

Notice that the ‘sprinkler’ table contains two distributions on $B$, one for the element $a \in A$ and one for $a' \in A$. But this is a channel, namely a channel $A \rightarrow B$. This is a crucial insight! We abbreviate this channel as $sp$, and define it as:

$$A \xrightarrow{sp} B \quad \text{with} \quad \begin{cases} sp(a) = \frac{1}{5}b + \frac{4}{5}b' \\ sp(a') = \frac{1}{5}b + \frac{4}{5}b' . \end{cases}$$

We read them as: if it’s winter, there is a 20% chance that the sprinkler is on, but if it’s not winter, there is 75% chance that the sprinkler is on.

Similarly, the ‘rain’ table corresponds to a channel:

$$A \xrightarrow{ra} C \quad \text{with} \quad \begin{cases} ra(a) = \frac{4}{5}c + \frac{1}{5}c' \\ ra(a') = \frac{1}{10}c + \frac{9}{10}c' . \end{cases}$$

Before continuing we can see that the formalisation (partial, so far) of the wetness Bayesian network in Figure 1.3 in terms of states and channels, already allows us to do something meaningful, namely state transformation $\gg$. Indeed, we can form distributions:

$$sp \gg wi \text{ on } B \quad \text{and} \quad ra \gg wi \text{ on } C .$$

These (transformed) distributions capture the derived probabilities that the
1.9. A Bayesian network example

sprinkler is on, and that it rains. Using Exercise 1.7.3(1) we get:

\[
(sp \gg wi)(b) = \sum_x sp(x)(b) \cdot wi(x) = sp(a)(b) \cdot wi(a) + sp(a')(b) \cdot wi(a') = \frac{1}{2} \cdot \frac{3}{5} + \frac{2}{5} \cdot \frac{3}{5} = \frac{21}{50}
\]

\[
(sp \gg wi)(b') = \sum_x sp(x)(b') \cdot wi(x) = sp(a')(b') \cdot wi(a) + sp(a')(b') \cdot wi(a') = \frac{2}{5} \cdot \frac{3}{5} + \frac{1}{2} \cdot \frac{2}{5} = \frac{34}{50}.
\]

Thus the overall distribution for the sprinkler (being on or not) is:

\[
sp \gg wi = \frac{21}{50} | b\rangle + \frac{34}{50} | b'\rangle.
\]

In a similar way one can compute the probability distribution for rain as:

\[
ra \gg wi = \frac{13}{25} | c\rangle + \frac{12}{25} | c'\rangle.
\]

Such distributions for non-initial nodes of a Bayesian network are called predictions. As will be shown here, they can be obtained via forward state transformation, following the structure of the network.

But first we still have to translate the upper two nodes of the network from Figure 1.3 into channels. In the conditional probability table for the ‘wet grass’ node we see 4 distributions on the set \(D\), one for each combination of elements from the sets \(B\) and \(C\). The table thus corresponds to a channel:

\[
B \times C \xrightarrow{wg} D
\]

with

\[
\begin{align*}
wg(b, c) &= \frac{19}{20} | d\rangle + \frac{1}{20} | d'\rangle \\
wg(b, c') &= \frac{9}{10} | d\rangle + \frac{1}{10} | d'\rangle \\
wg(b', c) &= \frac{4}{5} | d\rangle + \frac{1}{5} | d'\rangle \\
wg(b', c') &= 1 | d'\rangle.
\end{align*}
\]

Finally, the table for the ‘slippery road’ table gives:

\[
C \xrightarrow{sr} E
\]

with

\[
\begin{align*}
sr(c) &= \frac{7}{10} | e\rangle + \frac{3}{10} | e'\rangle \\
sr(c') &= 1 | e'\rangle.
\end{align*}
\]

We illustrate how to obtain predictions for ‘rain’ and for ‘slippery road’. We start from the latter. Looking at the network in Figure 1.3 we see that there are two arrows between the initial node ‘winter’ and our node of interest ‘slippery road’. This means that we have to do two state successive transformations, giving:

\[
(sr \circ ra) \gg wi = sr \gg (ra \gg wi) = sr \gg (\frac{12}{25} | c\rangle + \frac{12}{25} | c'\rangle) = \frac{91}{250} | e\rangle + \frac{159}{250} | e'\rangle.
\]

The first equation follows from Lemma 1.7.3(3). The second one involves elementary calculations, where we can use the distribution \(ra \gg wi\) that we calculated earlier.
Getting the predicted wet grass probability requires some care. Inspection of the network in Figure 1.3 is of some help, but leads to some ambiguity — see below. One might be tempted to form the parallel product $\otimes$ of the predicted distributions for sprinkler and rain, and do state transformation on this product along the wet grass channel $wg$, as in:

$$\textstyle wg \gg ((sp \gg wi) \otimes (ra \gg wi)).$$

But this is wrong, since the winter probabilities are now not used consistently, see the different outcomes in the calculations (1.12) and (1.13). The correct way to obtain the wet grass prediction involves copying the winter state, via the copy channel $\Delta$, see:

$$\textstyle (wg \circ (sp \otimes ra) \circ \Delta) \gg wi = wg \gg ((sp \otimes ra) \gg (\Delta \gg wi)) = \frac{1399}{2000}|d\rangle + \frac{601}{2000}|d\perp\rangle.$$

Such calculations are laborious, but essentially straightforward. We shall do this in one in detail, just to see how it works. Especially, it becomes clear that all summations are automatically done at the right place. We proceed in two steps, where for each step we only elaborate the first case.

$$(sp \otimes ra) \gg (\Delta \gg wi))(b, c)$$

$$\textstyle = \sum_{x,y}(sp \otimes ra)(x,y)(b, c) \cdot (\Delta \gg wi)(x,y)$$

$$\textstyle = \sum_{x,y}(sp(x) \otimes ra(y))(b, c) \cdot (\sum_{z}(\Delta(z)(x,y) \cdot wi(z))(x,y)$$

$$\textstyle = sp(a)(b) \cdot ra(a)(c) \cdot \frac{3}{5} + sp(a')(b) \cdot ra(a')(c) \cdot \frac{2}{5}$$

$$\textstyle = \frac{63}{500}$$

$$(sp \otimes ra) \gg (\Delta \gg wi))(b, c')$$

$$\textstyle = \frac{17}{500}$$

$$(sp \otimes ra) \gg (\Delta \gg wi))(b', c)$$

$$\textstyle = \frac{197}{500}$$

$$(sp \otimes ra) \gg (\Delta \gg wi))(b', c')$$

$$\textstyle = \frac{93}{500}$$

We conclude that:

$$sp \otimes ra \gg (\Delta \gg wi) = \frac{63}{500}(b, c) + \frac{147}{500}(b, c') + \frac{197}{500}(b', c) + \frac{93}{500}(b', c').$$
This distribution is used in the next step:

\[
\begin{align*}
&\left( wg \gg ((sp \otimes ra) \gg (\Delta \gg wi))(d) \\
&= \sum_{x,y} wg(x,y)(d) \cdot ((sp \otimes ra) \gg (\Delta \gg wi))(x,y) \\
&= wg(b,c)(d) \cdot \frac{63}{500} + wg(b,c')(d) \cdot \frac{147}{500} \\
&\quad + wg(b^*, c)(d) \cdot \frac{197}{500} + wg(b^*, c')(d) \cdot \frac{93}{500} \\
&= \frac{19}{25} \cdot \frac{63}{500} + \frac{9}{100} \cdot \frac{147}{500} + \frac{4}{5} \cdot \frac{197}{500} + 0 \cdot \frac{93}{500} \\
&= \frac{601}{2000}.
\end{align*}
\]

We have thus shown that:

\[
wg \gg ((sp \otimes ra) \gg (\Delta \gg wi))(d^+) = \frac{601}{2000}.
\]

### 1.9.1 Redrawing Bayesian networks

We have illustrated how prediction computations for Bayesian networks can be done, basically by following the graph structure and translating it into suitable sequential and parallel compositions (\(\circ\) and \(\otimes\)) of channels. The match between the graph and the computation is not perfect, and requires some care, especially wrt. copying. Since we have a solid semantics, we like to use it in order to improve the network drawing and achieve a better match between the underlying mathematical operations and the picture. Therefore we prefer to draw Bayesian networks slightly differently, making some minor changes:

- copying is written explicitly, for instance as \(\bigoplus\) in the binary case; in general one can have \(n\)-ary copying, for \(n \geq 2\);
- the relevant sets/types — like \(A, B, C, D, E\) — are not included in the nodes, but are associated with the arrows (wires) between the nodes;
- final nodes have outgoing arrows, labeled with their type.

The original Bayesian network in Figure 1.3 is then changed according to these points in Figure 1.4. In this way the nodes are clearly recognizable as channels, of type \(A_1 \times \cdots \times A_n \to B\), where \(A_1, \ldots, A_n\) are the types on the incoming wires, and \(B\) is the type of the outgoing wire. Initial nodes have no incoming wires, which formally leads to a channel \(1 \to B\), where 1 is the empty product. As we have seen, such channels \(1 \to B\) correspond to distributions/states on \(B\). In the adapted diagram one easily forms sequential and parallel compositions of channels, see the exercise below.
Exercises

1.9.1 In [20, §6.2] the (predicted) joint distribution on $D \times E$ that arises from the Bayesian network example in this section is represented as a table. It translates into:

$$\begin{array}{c|c|c|c|}
\text{winter} & \text{rain} & \text{sprinkler} & \text{slippery road} \\
\hline
d, e & \frac{30,443}{100,000} & \frac{39,507}{100,000} & \frac{5,957}{100,000} \\
\hline
d, \neg e & \frac{24,093}{100,000} & \frac{100,000}{100,000} & \frac{100,000}{100,000}
\end{array}$$

Following the structure of the diagram in Figure 1.4, it is obtained in the present setting as:

$$(\text{wg} \otimes \text{sr}) \circ (\text{id} \otimes \Delta) \circ (\text{sp} \otimes \text{ra}) \circ \Delta \gg wi$$

$$(\text{wg} \otimes \text{sr}) \gg ((\text{id} \otimes \Delta) \gg ((\text{sp} \otimes \text{ra}) \gg (\Delta \gg wi))).$$

Perform the calculations and check that this expression equals the above distribution.

Readers with access to [20] may wish to compare the different calculation methods, using sequential and parallel composition of channels — as here — or using multiplications of tables — as in [20].

1.10 The role of category theory

The previous sections have high-lighted several structural properties of, and similarities between, the collection types list, powerset, multiset and distribu-
1.10. The role of category theory

By now readers may ask: what is the underlying structure? Surely someone must have axiomatised what makes all of this work!

Indeed, this is called category theory. It provides a foundational language for mathematics, which was first formulated in the 1950s by Saunders MacLane and Samuel Eilenberg (see the first overview book [68]). Category theory focuses on the structural aspects of mathematics and shows that many mathematical constructions have the same underlying structure. It brings forward many similarities between different areas (see e.g. [69]). Category theory has become very useful in (theoretical) computer science too, since it involves a clear distinction between specification and implementation, see books like [4, 6, 65, 83]. We refer to those sources for more information.

The role of category theory in capturing the mathematical essentials and establishing connections also applies to probability theory. William Lawvere, another founding father of the area, first worked in this direction. Lawvere published little himself, but this line of work was picked up, extended and published by his PhD student Giry, whose name continues in the ‘Giry monad’ $\mathcal{G}$ of continuous probability distributions, see Section ?? of the first overview book [68]. The precise source of the distribution monad $\mathcal{D}$ for discrete probability theory, introduced in Section 1.5, is less clear, but it can be regarded as the discrete version of $\mathcal{G}$. Probabilistic automata have been studied in categorical terms as coalgebras, see e.g. [93] and [42] for general background information on coalgebra. There is a recent interest in more foundational, semantically oriented studies in probability theory, through the rise of probabilistic programming languages [34] and probabilistic Bayesian reasoning [55]. Probabilistic methods have received wider attention for instance, via the current interest in data analytics (see the essay [2]), in quantum probability [77], [16] and in cognition theory [39], [92].

Readers who know category theory will have recognised its implicit use in earlier sections. For readers who are not familiar (yet) with category theory, some basic concepts will be explained informally in this section. This is in no way a serious introduction to the area. The remainder of this book will continue to make implicit use of category theory, but will make this usage increasingly explicit. Hence it is useful to know the basic concepts of category, functor, natural transformation and monad. Category theory is sometimes seen as a difficult area to get into. But our experience is that it is easiest to learn category theory by recognising its concepts in constructions that you already know. That’s why this chapter started with concretely descriptions of various collections and their use in channels. For more solid expositions of category theory we refer to the sources listed above.
Chapter 1. Collections and Channels

1.10.1 Categories

A category is a mathematical structure given by a collection of ‘objects’ with ‘morphisms’ between them. The requirements are that these morphisms are closed under (associative) composition and that there is an identity morphism on each object. Morphisms are also called ‘maps’, and are written as \( f : X \rightarrow Y \), where \( X, Y \) are objects and \( f \) is a homomorphism from \( X \) to \( Y \). It is tempting to think of morphisms in a category as actual functions, but there are plenty of examples where this is not the case.

A category is like an abstract context of discourse, giving a setting in which one is working, with properties of that setting depending on the category at hand. We shall give a number of examples.

1. There is the category \( \text{Sets} \), whose objects are sets and whose morphisms are ordinary functions between them. This is a standard example.

2. One can also restrict to finite sets as objects, in the category \( \text{FinSets} \), with functions between them. This category is more restrictive, since for instance it contains objects \( n = \{0, 1, \ldots, n - 1\} \) for each \( n \in \mathbb{N} \), but not \( \mathbb{N} \) itself. Also, in \( \text{Sets} \) can take arbitrary products \( \prod_{i \in I} X_i \) of objects \( X_i \), over arbitrary index sets \( I \), whereas in \( \text{FinSets} \) only finite products exist. Hence \( \text{FinSets} \) is a more restrictive world.

3. Monoids and monoid maps have been mentioned in Definition 1.2.1. They can be organised in a category \( \text{Mon} \), whose objects are monoids, and whose homomorphisms are monoid maps. We now have to check that monoid maps are closed under composition and that identity functions are monoid maps; this is easy. Many mathematical structures can be organised into categories in this way, where the morphisms preserve the relevant structure. For instance, one can form a category \( \text{PoSets} \), with partially ordered sets (posets) as objects, and monotone functions between them as morphisms (also closed under composition, with identity).

4. For \( T \in \{L, P, M, D\} \) we can form the category \( \text{Chan}(T) \). Its objects are arbitrary sets \( X \), but its morphisms \( X \) to \( Y \) are \( T \)-channels, \( X \rightarrow T(Y) \), written as \( X \rightarrow Y \). We have already seen that channels are closed under composition \( \circ \) and have \( \text{unit} \) as identity, see Lemma 1.7.3. We can now say that \( \text{Chan}(T) \) is a category. In the sequel we often write \( \text{Chan} \) for \( \text{Chan}(D) \), since that is our most important example.

These categories of channels form good examples of the idea that a category forms a universe of discourse. For instance, in \( \text{Chan}(P) \) we are in the world of non-deterministic computation, whereas \( \text{Chan} = \text{Chan}(D) \) is the world of probabilistic computation.
1.10. The role of category theory

We will encounter several more examples of categories later on in the book. Occasionally, the following construction will be used. Given a category $\mathcal{C}$, a new ‘opposite’ category $\mathcal{C}^{\text{op}}$ is formed. It has the same objects as $\mathcal{C}$, but its morphisms are reversed. Thus $f: Y \to X$ in $\mathcal{C}^{\text{op}}$ means $f: X \to Y$ in $\mathcal{C}$.

1.10.2 Functors

Category theorists like abstraction, hence the question: if categories are so important, then why not organise them as objects themselves in a superlarge category $\text{Cat}$, with morphisms between them preserving the relevant structure? The latter morphisms between categories are called ‘functors’. More precisely, given categories $\mathcal{C}$ and $\mathcal{D}$, a functor $F: \mathcal{C} \to \mathcal{D}$ between them consists of two mappings, both written $F$, sending an object $X$ in $\mathcal{C}$ to an object $F(X)$ in $\mathcal{D}$, and a morphism $f: X \to Y$ in $\mathcal{C}$ to a morphism $F(f): F(X) \to F(Y)$ in $\mathcal{D}$. This mapping $F$ should preserve composition and identities, as in: $F(g \circ f) = F(g) \circ F(f)$ and $F(\text{id}_X) = \text{id}_{F(X)}$.

Earlier we have already called some operations ‘functorial’ for the fact that they preserve composition and identities. We can now be a bit more precise.

1 Each $T \in \{L, P, P_{\text{fin}}, M, M_{\ast}, N, N_{\ast}, D, D_{\infty}\}$ is a functor $T: \text{Sets} \to \text{Sets}$. This has been described in the beginning of each of the sections 1.2 – 1.5.

2 Taking lists is also a functor $L: \text{Sets} \to \text{Mon}$. This is in essence that content of Lemma 1.2.2. One can also view $P, P_{\text{fin}}$ and $M$ as functors $\text{Sets} \to \text{Mon}$, see Lemmas 1.3.1 and 1.4.2. Even more, one can describe $P, P_{\text{fin}}$ as a functor $\text{Sets} \to \text{PoSets}$, by considering each set of subsets $P(X)$ and $P_{\text{fin}}(X)$ with its subset relation $\subseteq$ as partial order. In order to verify this claim one has to check that $P(f): P(X) \to P(Y)$ is a morphism of posets, that it, forms a monotone function. But that’s easy.

3 There is also a functor $J: \text{Sets} \to \text{Chan}(T)$, for each $T$. It is the identity on sets/objects: $J(X) := X$. But it sends a function $f: X \to Y$ to the channel $J(f) := \langle f \rangle = \text{unit} \circ f: X \to Y$. We have seen, in Lemma 1.7.3(4), that $J(g \circ f) = J(g) \circ J(f)$ and that $J(\text{id}) = \text{id}$, where the latter identity $\text{id}$ is $\text{unit}$ in the category $\text{Chan}(T)$. This functor $J$ shows how to embed the world of ordinary computations (functions) into the world of computations of type $T$ (channels).

1.10.3 Natural transformations

Let’s move one further step up the abstraction ladder and look at morphisms between functors. They are called natural transformations. We have already
seen examples of those as well. Given two functors \( F, G : C \to D \), a natural transformation \( \alpha : F \to G \) is a collection of maps \( \alpha_X : F(X) \to G(X) \) in \( D \), indexed by objects \( X \) in \( C \). Naturality means that \( \alpha \) works in the same way on all objects and is expressed as follows: for each morphism \( f : X \to Y \) in \( C \), the following rectangle in \( D \) commutes.

\[
\begin{array}{c}
F(X) \xrightarrow{\alpha_X} G(X) \\
\downarrow_{F(f)} \quad \downarrow_{G(f)} \\
F(Y) \xrightarrow{\alpha_Y} G(Y)
\end{array}
\]

We briefly review some of the examples of natural transformations that we have seen.

1. The various support maps can now be described as natural transformations \( \text{supp} : \mathcal{L} \to \mathcal{P}_{\text{fin}}, \text{supp} : \mathcal{M} \to \mathcal{P}_{\text{fin}} \) and \( \text{supp} : \mathcal{D} \to \mathcal{P}_{\text{fin}} \), see the overview diagram 1.5.

2. The naturality of the frequentist learning map \( \text{Flrn} : \mathcal{M} \to \mathcal{D} \) has occurred earlier as an explicit result, see Lemma 1.6.2.

3. For each \( T \in \{ \mathcal{L}, \mathcal{P}, \mathcal{M}, \mathcal{D} \} \) we have described maps \( \text{unit} : X \to T(X) \) and \( \text{flat} : T(T(X)) \to T(X) \) and have seen naturality results about them. We can now state more precisely that they are natural transformations \( \text{unit} : \text{id} \to T \) and \( \text{flat} : (T \circ T) \to T \). Here we have used \( \text{id} \) as the identity functor \( \text{Sets} \to \text{Sets} \), and \( T \circ T \) as the composite of \( T \) with itself, also as a functor \( \text{Sets} \to \text{Sets} \).

### Monads

A monad on a category \( C \) is a functor \( T : C \to C \) that comes with two natural transformations \( \text{unit} : \text{id} \to T \) and \( \text{flat} : (T \circ T) \to T \) satisfying \( \text{flat} \circ \text{unit} = \text{id} = \text{flat} \circ T(\text{unit}) \) and \( \text{flat} \circ \text{flat} = \text{flat} \circ T(\text{flat}) \). All the collection functors \( \mathcal{L}, \mathcal{P}, \mathcal{P}_{\text{fin}}, \mathcal{M}, \mathcal{M}_{\text{fin}}, \mathcal{N}, \mathcal{N}_{\text{fin}}, \mathcal{D}, \mathcal{D}_{\text{fin}} \) for which we have seen so far are monads, see e.g. Lemma 1.2.3, 1.3.2 of 1.4.3. For each monad \( T \) we can form a category \( \text{Chan}(T) \) of \( T \)-channels, that capture computations of type \( T \). In category theory this is called the Kleisli category of \( T \). Monads have become popular in functional programming \( \text{\cite{monads}} \) as mechanisms for including special effects (e.g., for input-output, writing, side-effects, continuations) into a functional programming language.\footnote{See the online overview https://wiki.haskell.org/Monad_tutorials_timeline} The structure of probabilistic computation is
also given by monads, namely by the discrete distribution monads $\mathcal{D}, \mathcal{D}_\infty$, and by the continuous distribution monad $\mathcal{G}$.

### Exercises

1.10.1 We have seen the functor $J : \text{Sets} \to \text{Chan}(T)$. Check that there is also a functor $\text{Chan}(T) \to \text{Sets}$ in the opposite direction, which is $X \mapsto T(X)$ on objects, and $c \mapsto c \gg (\_)$ on morphisms. Check explicitly that composition is preserved, and find the earlier result that stated that fact implicitly.

1.10.2 Recall from Exercise 1.5.7 the subset $\mathcal{N}[K](X) \subseteq \mathcal{M}(X)$ of natural multisets with $K$ elements. Prove that $\mathcal{N}[K]$ is a functor $\text{Sets} \to \text{Sets}$.
Predicates and Observables

After a general introduction in the previous chapter, the focus now lies firmly on probability. We have seen (discrete probability) distributions \( \omega = \sum_i r_i |x_i \) in \( D(X) \) and channels \( X \rightarrow Y \), as functions \( X \rightarrow D(Y) \), describing probabilistic states and computations. This section develops the tools to reason about such states and computations, via what we call observables. They are functions from \( X \) to (a subset of) the real numbers \( \mathbb{R} \) that associate some ‘observable’ numerical information with an element \( x \in X \) of the sample space \( X \) of a distribution. The following table gives an overview, where \( X \) is a set (used as sample space).

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>observable / utility function</td>
<td>( X \rightarrow \mathbb{R} )</td>
</tr>
<tr>
<td>factor / potential</td>
<td>( X \rightarrow \mathbb{R}_{\geq 0} )</td>
</tr>
<tr>
<td>(fuzzy) predicate / (soft/uncertain) evidence</td>
<td>( X \rightarrow [0, 1] )</td>
</tr>
<tr>
<td>sharp predicate / event</td>
<td>( X \rightarrow {0, 1} )</td>
</tr>
</tbody>
</table>

We shall use the term of ‘observable’ as generic expression for the entries in this table. A function \( X \rightarrow \mathbb{R} \) is thus the most general type of observable, and a sharp predicate is the most specific one. Predicates are the most appropriate observable for probabilistic logical reasoning. Often attention is restricted to subsets \( U \subseteq X \) as predicates (or events [57]), but here the fuzzy versions \( X \rightarrow [0, 1] \) are the default. A technical reason for this choice is that these fuzzy predicates are closed under predicate transformation, and the sharp predicates are not, see below for details.

This chapter starts with the definition of what can be seen as probabilistic truth \( \omega \models p \), namely the validity of an observable \( p \) in a state \( \omega \). We shall see
2.1 Validity

that many basic concepts can be defined in terms of validity, including mean, average, entropy, distance, (co)variance. The algebraic, logical and categorical structure of the various observables in Table 2.1 will be investigated in Section 2.2. For a factor (or predicate or event) $p$ and a state $\omega$, both on the same sample space, one can form a new updated (conditioned) state $\omega|_p$. It is the state $\omega$ which is updated in the light of evidence $p$. This updating satisfies various properties, including Bayes’ rule. A theme that will run through this book is that probabilistic updating (conditioning) has ‘crossover’ influence. This means that if we have a joint state on a product sample space, then updating in one component typically changes the state in the other component (the marginal). This crossover influence is characteristic for a probabilistic setting and depends on correlation between the two components. As we shall see later, channels play an important role in this phenomenon.

In the previous chapter we have seen that a state $\omega$ on the domain $X$ of a channel $c: X \to Y$ can be transformed into a state $c \gg \omega$ on the channel’s codomain $Y$. Analogous to such state transformation $\gg$ there is also observable transformation $\prec$, acting in the opposite direction: for an observable $q$ on the codomain $Y$ of a channel $c: X \to Y$, there is a transformed observable $c \gg q$ on $X$. When $\prec$ is applied to predicates, it is called predicate transformation. It is a basic operation in programming logics. The power of conditioning becomes apparent when it is combined with transformation, especially for inference in probabilistic reasoning. We shall distinguish two forms. Forward inference involves conditioning followed by state transformation. Backward inference involves observable transformation followed by conditioning. We illustrate the usefulness of these inference techniques in many examples in Sections 2.5 and 2.6, but also in Section 2.7 for Bayesian networks. The last three sections of this chapter use validity $\equiv$ for the definition of distances — both between states and between predicates — in a dual manner, and for a systematic account of (co)variance and correlation.

2.1 Validity

This section introduces the basic facts and terminology for observables, as described in Table 2.1. Recall that we write $Y^X$ for the set of functions from $X$ to $Y$. We will use notations:

- $\text{Obs}(X) := \mathbb{R}^X$ for the set of observables on $X$;
- $\text{Fact}(X) := (\mathbb{R}_{\geq 0})^X$ for the factors on $X$;
- $\text{Pred}(X) := [0, 1]^X$ for the predicates on $X$;
Chapter 2. Predicates and Observables

- $\text{SPred}(X) := \{0, 1\}^X$ for the sharp predicates (events) on $X$.

There are inclusions:

$$\text{SPred}(X) \subseteq \text{Pred}(X) \subseteq \text{Fact}(X) \subseteq \text{Obs}(X).$$

The first set $\text{SPred}(X) = \{0, 1\}^X$ of sharp predicates can be identified with the powerset $\mathcal{P}(X)$ of subsets of $X$, see below. We first define some special observables.

**Definition 2.1.1.** Let $X$ be an arbitrary set.

1. For a subset $U \subseteq X$ we write $1_U : X \to \{0, 1\}$ for the characteristic function of $U$, defined as:

   $$1_U(x) = \begin{cases} 1 & \text{for } x \in U \\ 0 & \text{otherwise.} \end{cases}$$

   This function $1_U : X \to [0, 1]$ is the (sharp) predicate associated with the subset $U \subseteq X$.

2. We use special notation for two extreme cases $U = X$ and $U = \emptyset$, giving the *truth* predicate $1 : X \to [0, 1]$ and the *falsity* predicate $0 : X \to [0, 1]$ on $X$. Explicitly:

   $$1 = 1_X \text{ and } 0 = 1_\emptyset \text{ so that } 1(x) = 1 \text{ and } 0(x) = 0,$$

   for all $x \in X$.

3. For a singleton subset $\{x\}$ we simply write $1_x$ for $1_{\{x\}}$. Such functions $1_x : X \to [0, 1]$ are also called *point predicates*, where the element $x \in X$ is seen as a point.

4. There is a (sharp) equality predicate $\text{Eq} : X \times X \to [0, 1]$ defined in the obvious way as:

   $$\text{Eq}(x, x') := \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise.} \end{cases}$$

5. If the set $X$ can be mapped into $\mathbb{R}$ in an obvious manner, we write this inclusion function as $\text{incl}_X : X \leftrightarrow \mathbb{R}$ and may consider it as a random variable on $X$. This applies for instance if $X = \mathbb{N} = \{0, 1, \ldots, n-1\}$.

   This mapping $U \mapsto 1_U$ forms the isomorphism $\mathcal{P}(X) \cong \{0, 1\}^X$ that we mentioned just before Definition 2.1.1. In analogy with point (3) we sometimes call a state of the form $\text{unit}(x) = 1|\langle x \rangle$ a *point state*.

**Definition 2.1.2.** Let $X$ be a set with an observable $p : X \to \mathbb{R}$ on $X$. 

64
2.1. Validity

1 For a distribution/state \( \omega = \sum_i r_i |x_i \rangle \) on \( X \) we define its validity \( \omega \models p \) as:
\[
\omega \models p := \sum_i r_i \cdot p(x) = \sum_{x \in X} \omega(x) \cdot p(x).
\]

Elsewhere, this validity is often called expected value.

2 If \( X \) is a finite set one can define the average \( \text{avg}(p) \) of the observable \( p \) as its validity in the uniform state \( \upsilon_X \) on \( X \), i.e.,

\[
\text{avg}(p) := \upsilon_X \models p = \sum_{x \in X} \frac{p(x)}{n}\text{,}
\]

where \( n \) is the number of elements of \( X \).

The validity \( \omega \models p \) is non-negative (is in \( \mathbb{R} \geq 0 \)) if \( p \) is a factor and lies in the unit interval \([0, 1] \) if \( p \) is a predicate (whether sharp or not). The fact that the multiplicities \( r_i \) in a distribution \( \omega = \sum_i r_i |x_i \rangle \) add up to one means that the validity \( \omega \models 1 \) is one. Notice that for a point predicate one has \( \omega \models 1_x = \omega(x) \) and similarly, for a point state, \( \text{unit}(x) \models p = p(x) \).

As an aside, we typically do not write brackets in equations like \( (\omega \models p) = a \), but use the convention that \( | \) has higher precedence than \( = \), so that the equation can simply be written as \( \omega \models p = a \). Similarly, one can have validity \( c \gg \omega \models p \) in a transformed state, which should be read as \( (c \gg \omega) \models p \).

We add two more notions that are defined in terms of validity.

**Definition 2.1.3.** Let \( \omega \) be a distribution/state on a set \( X \).

1 Write \( I(\omega) : X \to \mathbb{R}_{\geq 0} \) for the information content of the distribution \( \omega \), defined as:
\[
I(\omega)(x) := -\log(\omega(x)) \quad \text{where } \log = \log_2 \text{ is the logarithm with base } 2.
\]

The Shannon entropy \( H(\omega) \) of \( \omega \) is the validity of the factor \( I(\omega) \):
\[
H(\omega) := \omega \models I(\omega) = -\sum_x \omega(x) \cdot \log(\omega(x)).
\]

(Sometimes we also use the natural logarithm \( \ln \) instead of \( \log \).)

2 In the presence of a map \( \text{incl}_X : X \hookrightarrow \mathbb{R} \), one can define the mean \( \text{mean}(\omega) \) of the distribution \( \omega \) on \( X \) as the validity of \( \text{incl}_X \), considered as a random variable:
\[
\text{mean}(\omega) := \omega \models \text{incl}_X = \sum_{x \in X} \omega(x) \cdot x.
\]

**Example 2.1.4.** 1 Let \( \text{flip}(0.3) = 0.3 |1 \rangle + 0.7 |0 \rangle \) be a biased coin. Suppose there is a game where you can throw the coin and win \( €100 \) if head (1) comes up, but you lose \( €50 \) if the outcome is tail (0). Is it a good idea to play the game?

The possible gain can be formalised as a random variable \( v : \{0, 1\} \to \mathbb{R} \)
66 Chapter 2. Predicates and Observables

with \( v(0) = -50 \) and \( v(1) = 100 \). We get an answer to the above question by computing the validity:

\[
\text{flip}(0.3) \models v = \sum_x \text{flip}(0.3)(x) \cdot v(x) \\
= \text{flip}(0.3)(0) \cdot v(0) + \text{flip}(0.3)(1) \cdot v(1) \\
= 0.7 \cdot -50 + 0.3 \cdot 100 = -35 + 30 = -5.
\]

Hence it is wiser not to play.

2 Write \( \text{pips} \) for the set \( \{1, 2, 3, 4, 5, 6\} \), considered as a subset of \( \mathbb{R} \), via the map \( \text{incl}: \text{pips} \hookrightarrow \mathbb{R} \), which is in its turn is seen as an observable. The average of the latter is its validity \( \text{dice} \models \text{incl} \) for the (uniform) fair dice state \( \text{dice} = v_{\text{pips}} = \frac{1}{6}|1| + \frac{1}{6}|2| + \frac{1}{6}|3| + \frac{1}{6}|4| + \frac{1}{6}|5| + \frac{1}{6}|6| \). This average is 3.5. It is at the same time the expected outcome after a throw of the dice.

3 Suppose that we claim that in a throw of a (fair) dice the outcome is even. How likely is this claim? We formalise it as a (sharp) predicate \( e: \text{pips} \rightarrow [0,1] \) with \( e(1) = e(3) = e(5) = 0 \) and \( e(2) = e(4) = e(6) = 1 \). Then, as expected:

\[
\text{dice} \models e = \sum_x \text{dice}(x) \cdot e(x) \\
= \frac{1}{6} \cdot 0 + \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 0 + \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 0 + \frac{1}{6} \cdot 1 = \frac{1}{2}.
\]

Now consider a more subtle claim that the even pips are more likely than the odd ones, where the precise likelihoods are described by the (non-sharp, fuzzy) predicate \( p: \text{pips} \rightarrow [0,1] \) with:

\[
p(1) = \frac{1}{10} \quad p(2) = \frac{9}{10} \quad p(3) = \frac{3}{10} \quad p(4) = \frac{8}{10} \quad p(5) = \frac{2}{10} \quad p(6) = \frac{7}{10}.
\]

This new claim \( p \) happens to be equally probable as the even claim \( e \), since:

\[
\text{dice} \models p = \frac{1}{6} \cdot \frac{1}{10} + \frac{1}{6} \cdot \frac{9}{10} + \frac{1}{6} \cdot \frac{8}{10} + \frac{1}{6} \cdot \frac{3}{10} + \frac{1}{6} \cdot \frac{2}{10} + \frac{1}{6} \cdot \frac{7}{10} \\
= 1 + 9 + 3 + 8 + 2 + 7 = \frac{30}{60} = \frac{1}{2}.
\]

4 Recall the binomial distribution \( \text{binom}[K](r) \) on \( \{0, 1, \ldots, K\} \) from Example 1.5.1 (2). There is an inclusion function \( [0, 1, \ldots, K] \hookrightarrow \mathbb{R} \) that allows
us to compute the mean of the binomial distribution as:

\[
\text{mean}(\text{binom}[K](r)) = \sum_{0 \leq k \leq K} \binom{K}{k} \cdot r^k \cdot (1 - r)^{K-k} \cdot k
\]

\[
= \sum_{0 \leq k \leq K} \frac{k \cdot K!}{k! \cdot (K - k)!} \cdot r^k \cdot (1 - r)^{K-k}
\]

since the term for \( k = 0 \) can be dropped

\[
= \sum_{1 \leq k \leq K} \frac{K \cdot (K - 1)!}{(k - 1)! \cdot ((K - 1) - (k - 1))!} \cdot r^k \cdot (1 - r)^{K-1-k}
\]

\[
= K \cdot r \cdot \sum_{0 \leq j \leq K-1} \frac{(K - 1)!}{j! \cdot (K - 1 - j)!} \cdot r^j \cdot (1 - r)^{K-1-j}
\]

\[
= K \cdot r \cdot \sum_{0 \leq j \leq K-1} \binom{K-1}{j}(r)(j)
\]

\[
= K \cdot r.
\]

In the two plots of binomial distributions \( \text{binom}[15](\frac{1}{3}) \) and \( \text{binom}[15](\frac{2}{3}) \) in Figure 1.2 (on page 25) one can see that the associated means \( \frac{15}{3} = 5 \) and \( \frac{45}{3} = 15.25 \) make sense.

We elaborate multinomial distributions in a separate example.

**Example 2.1.5.** In Exercise [15.7] we have suggested generalisations of the binomial distribution to the multinomial case. Here we shall describe this generalisation in detail, and describe the associated mean value calculations.

Fix a natural number \( K > 0 \). Recall that we write \( \mathcal{N}[K](X) \subseteq M(X) \) for the subset of multisets with \( K \) elements. For \( \vec{k} = (k_1, \ldots, k_m) \in \mathcal{N}[K](m) \) we have \( \sum_i k_i = K \). One writes:

\[
\binom{K}{\vec{k}} := \frac{K!}{k_1! \cdots k_m!} = \frac{K}{\prod_i k_i!}.
\]

For \( \vec{x} = (x_1, \ldots, x_m) \in \mathcal{D}(m) \), so that \( \sum_i x_i = 1 \), we write:

\[
\vec{x}^{\vec{k}} := x_1^{k_1} \cdots x_m^{k_m} = \prod_i x_i^{k_i}.
\]

We use these notations to define the multinomial map:

\[
\mathcal{D}(m) \xrightarrow{\text{multinom}[K]} \mathcal{D}(\mathcal{N}[K](m))
\]

67
as the following distribution on multisets:
\[
multinom[K](\mathbf{x}) := \sum_{\mathbf{k} \in \mathbb{N}^m} \left( \begin{array}{c} K \\ \mathbf{k} \end{array} \right) \mathbf{x}^\mathbf{k}.
\]

This \( \text{multinom}[K](\mathbf{x}) \) is a distribution on multisets: its multiplicities add up to one by the multinomial theorem,
\[
\sum_{\mathbf{k} \in \mathbb{N}^m} \left( \begin{array}{c} K \\ \mathbf{k} \end{array} \right) \mathbf{x}^\mathbf{k} = (x_1 + \cdots + x_n)^K = 1^K = 1.
\]

Our aim is to calculate the mean of this distribution \( \text{multinom}[K](\mathbf{x}) \). We do so via the following auxiliary result:
\[
\sum_{\mathbf{k} \in \mathbb{N}^m} k_i \left( \begin{array}{c} K \\ \mathbf{k} \end{array} \right) \mathbf{x}^\mathbf{k} = Kx_i. 
\tag{2.2}
\]

The proof involves some basic shifting:
\[
\sum_{\mathbf{k} \in \mathbb{N}^m} k_i \left( \begin{array}{c} K \\ \mathbf{k} \end{array} \right) \mathbf{x}^\mathbf{k} = \sum_{\mathbf{k} \in \mathbb{N}^m, k_i > 0} k_i \left( \begin{array}{c} K \\ k_i \end{array} \right) x_1^{k_1} \cdots x_i^{k_i} \cdots x_m^{k_m}
\]
\[
= \sum_{\mathbf{k} \in \mathbb{N}^m, k_i > 0} k_i (K-1)! k_i \cdots (k_i-1)! \cdots k_m! x_1^{k_i} \cdots x_i^{k_i-1} \cdots x_m^{k_m}
\]
\[
= Kx_i \sum_{\mathbf{k} \in \mathbb{N}^{K-1}(m)} (K-1)! k_1 \cdots k_i \cdots k_m! x_1^{k_1} \cdots x_i^{k_i-1} \cdots x_m^{k_m}
\]
\[
= Kx_i.
\]

We can use inclusions \( \mathbb{N}^K(m) \subseteq M(m) \subseteq (\mathbb{R}_{\geq 0})^m \) to see multisets on \( m \) as \( m \)-tuples \( \mathbf{k} \in (\mathbb{R}_{\geq 0})^m \). Thus we have \( m \) projection factors \( \pi_i : \mathbb{N}^K(m) \rightarrow \mathbb{R}_{\geq 0} \), for which we can obtain the validity:
\[
\text{multinom}[K](\mathbf{x}) \models \pi_i = \sum_{\mathbf{k} \in \mathbb{N}^K(m)} \left( \begin{array}{c} K \\ \mathbf{k} \end{array} \right) \mathbf{x}^\mathbf{k} \pi_i \equiv Kx_i.
\]

The \( m \)-tuple of these validities is sometimes considered as the mean of the multinomial \( \text{multinom}[K](\mathbf{x}) \).

**Remark 2.1.6.** In the previous example we have considered a multiset as a factor. This works, since a multiset on \( X \) can be seen as function \( X \rightarrow \mathbb{R}_{\geq 0} \) with finite support. In the same way one can regard a distribution on \( X \), seen as a function \( X \rightarrow [0, 1] \) with finite support, as a predicate on \( X \). Thus there are inclusions:
\[
\mathcal{M}(X) \subseteq \text{Fact}(X) = (\mathbb{R}_{\geq 0})^X \quad \text{and} \quad \mathcal{D}(X) \subseteq \text{Pred}(X) = [0, 1]^X.
\]
2.1. Validity

The first inclusion \( \subseteq \) may actually be seen as an equality \( = \) when the set \( X \) is finite.

We are however reluctant to identify distributions with certain predicates, since they belong to different universes and have quite different algebraic properties. For instance, distributions are convex sets, whereas predicates are effect modules carrying a commutative monoid, see the next section. Keeping states and predicates apart is a matter of mathematically hygiene\(^1\). We have already seen state transformation \( \Rightarrow \) along a channel; it preserves the convex structure. Later on in this chapter we shall also see predicate (or: observable) transformation \( \ll \) along a channel, in opposite direction; this operation preserves the effect module structure on predicates.

Apart from these mathematical differences, states and predicates play entirely different roles and are understood in different ways: states play an ontological role and describe ‘states of affairs’; predicates are epistemological in nature, and describe evidence (what we know).

Whenever we make use of the above inclusions, we shall make this usage explicit.

2.1.1 Random variables

So far we have avoided discussing the concept of ‘random variable’. It turns out to be a difficult notion for people who start studying probability theory. One often encounters phrases like: let \( Y \) be a random variable with expectation \( E[Y] \). How should this be understood? We choose to interpret a random variable as a pair.

Definition 2.1.7. 1 A random variable on a sample space (set) \( X \) consists of two parts:

- a state \( \omega \in \mathcal{D}(X) \);
- an observable \( R: X \to \mathbb{R} \).

2 The probability mass function \( P[R = (-)] : \mathbb{R} \to [0,1] \) associated with the random variable \((\omega,R)\) is the image distribution on \( \mathbb{R} \) given by:

\[
P[R = (-)] := R \gg \omega = \mathcal{D}(R)(\omega).
\]

Hence:

\[
P[R = r] = (R \gg \omega)(r) = \omega \gg R \models 1_r.
\]

(We use the function \( R: X \to \mathbb{R} \) as deterministic channel in writing \( R \gg \omega \), see Lemma 1.7.3\(^2\).)

\(^1\) Moreover, in continuous probability there are no inclusions of states in predicates.
The expected value of a random variable \((\omega, R)\) is the mean:

\[
\text{mean}(R \gg \omega) = \omega \models R.
\]

The last equation holds since:

\[
\begin{align*}
\text{mean}(R \gg \omega) &= \sum_r (R \gg \omega)(r) \cdot r \\
&= \sum_r \mathcal{D}(R)(\omega)(r) \cdot r \\
&= \sum_{x \in R^{-1}(r)} \omega(x) \cdot r \\
&= \sum_x \omega(x) \cdot R(x) \\
&= \omega \models R.
\end{align*}
\]

The interpretation of a random variable as a pair, of a state \(\omega\) and an observable \(R\), makes sense in the light of the habit in traditional probability to keep the state implicit, and thus only talk about the observable \(R\). However, expressions like \(P[R = r]\) and \(E[R]\) only make sense in presence of a state on the underlying sample space of the observable \(R\). Here, we make this state explicit.

**Example 2.1.8.** An easy illustration is the expected value for the sum of two dice. In this situation we have an observable \(S : \text{pips} \times \text{pips} \to \mathbb{R}\), on the sample space \(\text{pips} = \{1, 2, 3, 4, 5, 6\}\), given by \(S(i, j) = i + j\). It forms a random variable together with the product \(\text{dice} \otimes \text{dice} \in \mathcal{D}(\text{pips} \times \text{pips})\) of the uniform distribution \(\text{dice} = v = \sum_{i \in \text{pips}} \frac{1}{6} |i\).

The expected value of the random variable \((\text{dice} \otimes \text{dice}, S)\) is, according to Definition 2.1.7 (3),

\[
\begin{align*}
\text{mean}(S \gg \text{dice} \otimes \text{dice}) &= \text{dice} \otimes \text{dice} \models S \\
&= \sum_{i,j \in \text{pips}} \text{dice} \otimes \text{dice}(i, j) \cdot S(i, j) \\
&= \sum_{i,j \in \text{pips}} \frac{1}{6} \cdot \frac{1}{6} \cdot (i + j) \\
&= \frac{\sum_{i,j \in \text{pips}} i + j}{36} = \frac{252}{36} = 7.
\end{align*}
\]

In the context of this book, and more broadly, in probabilistic programming, state transformation \(\gg\) plays an important role. Therefore we insist on writing states explicit, so that it is always clear in which state we are doing what. In mathematical contexts, the state/distribution is often the same (not transformed) and can be understood from the context — which justifies omitting it. That’s why we shall use \(\omega \models p\) in the sequel, instead of \(E[p]\).
2.2. The structure of observables

Exercises

2.1.1 Check that the average of the set \( n + 1 = \{0, 1, \ldots, n\} \), considered as random variable, is \( \frac{n}{2} \).

2.1.2 In Example 2.1.8 we have seen that \( \text{dice} \otimes \text{dice} \models S = 7 \), for the observable \( S : \text{pips} \times \text{pips} \to \mathbb{R} \) by \( S(x, y) = x + y \).

1. Now define \( T : \text{pips}^3 \to \mathbb{R} \) by \( T(x, y, z) = x + y + z \). Prove that \( \text{pips} \otimes \text{pips} \otimes \text{pips} \models T = \frac{31}{2} \).

2. Can you generalise and show that summing on \( \text{pips}^n \) yields validity \( \frac{7n^2}{2} \)?

2.1.3 Consider an arbitrary distribution \( \omega \in \mathcal{D}(X) \).

1. Check that for a function \( f : X \to Y \) and an observable \( q \in \text{Obs}(Y) \),

\[
q \gg \omega \models q = \omega \models q \circ f
\]

2. Observe that we get as special case, for an observable \( q : X \to \mathbb{R} \),

\[
q \gg \omega \models \text{id} = \omega \models q,
\]

where the identity function \( \text{id} : \mathbb{R} \to \mathbb{R} \) is used as observable.

2.1.4 The point of this section is to show that “learning after multinomial” is the identity, where “after” refers to composition of probabilistic channels.

1. Describe frequentist learning as channel \( \text{Flrn} : \mathcal{M}_c(m) \to m \) and multinomial as channel \( \text{multinom}[K] : \mathcal{D}(m) \to \mathcal{M}_c(m) \), for a fixed \( K > 0 \). Further, notice that the identity function \( \mathcal{D}(m) \to \mathcal{D}(m) \) forms a channel \( \text{id} : \mathcal{D}(m) \to m \).

2. Use (2.2) to prove that the following triangle of channels commutes:

\[
\begin{array}{ccc}
\mathcal{D}(m) & \xrightarrow{\text{multinom}[K]} & \mathcal{M}_c(m) \\
\downarrow{\text{id}} & & \downarrow{\text{Flrn}} \\
\mathcal{D}(m) & \xrightarrow{\mathcal{M}_c(m)} & m
\end{array}
\]

2.2 The structure of observables

This section describes the algebraic structures that the various types of observables have, without going too much into mathematical details. We don’t want to turn this section into a sequence of formal definitions. Hence we present the essentials and refer to the literature for details. We include the observation that mapping a set \( X \) to observables on \( X \) is functorial, in a suitable sense. This
will give rise to the notion of *weakening*. It plays the same (or dual) role for observables that marginalisation plays for states.

It turns out that our four types of observables are all (commutative) monoids, via multiplication / conjunction, but in different universes. The findings are summarised in the following table.

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>monoid in</th>
</tr>
</thead>
<tbody>
<tr>
<td>observable</td>
<td>$X \rightarrow \mathbb{R}$</td>
<td>ordered vector spaces</td>
</tr>
<tr>
<td>factor</td>
<td>$X \rightarrow \mathbb{R}_{\geq 0}$</td>
<td>ordered cones</td>
</tr>
<tr>
<td>predicate</td>
<td>$X \rightarrow [0, 1]$</td>
<td>effect modules</td>
</tr>
<tr>
<td>sharp predicate</td>
<td>$X \rightarrow {0, 1}$</td>
<td>join semilattices</td>
</tr>
</tbody>
</table>

The least well-known structures in this table are effect modules. They will thus be described in greatest detail, in Subsection 2.2.3.

### 2.2.1 Observables

Observables are $\mathbb{R}$-valued functions on a sample space which are often written as capitals $X, Y, \ldots$. Here, these letters are typically used for samples spaces themselves. We shall use letters $p, q, \ldots$ for observables in general, and in particular for predicates. In some settings one allows observables $X \rightarrow \mathbb{R}^n$ to the $n$-dimensional space of real numbers. Whenever needed, we shall use such maps as $n$-ary tuples $\langle p_1, \ldots, p_n \rangle: X \rightarrow \mathbb{R}^n$ of random variables $p_i: X \rightarrow \mathbb{R}$, see also Section 1.1.

Let's fix a set $X$, and consider the collection $\text{Obs}(X) = \mathbb{R}^X$ of observables on $X$. What structure does it have?

- Given two observables $p, q \in \text{Obs}(X)$, we can add them point-wise, giving $p + q \in \text{Obs}(X)$ via $(p + q)(x) = p(x) + q(x)$.
- Given an observable $p \in \text{Obs}(X)$ and a scalar $s \in \mathbb{R}$ we can form a ‘re-scaled’ observable $s \cdot p \in \text{Obs}(X)$ via $(s \cdot p)(x) = s \cdot p(x)$. In this way we get $-p = (-1) \cdot p$ and $0 = 0 \cdot p$ for any observable $p$, where $0 = 1_\emptyset \in \text{Obs}(X)$ is taken from Definition 2.1.1 (2).
- For observables $p, q \in \mathbb{R}^X$ we have a partial order $p \leq q$ defined by: $p(x) \leq q(x)$ for all $x \in X$.

Together these operations of sum $+$ and scalar multiplication $\cdot$ make $\mathbb{R}^X$ into a
vector space over the real numbers, since $+$ and $\cdot$ satisfy the appropriate axioms of vector spaces. Moreover, this is an ordered vector space by the third bullet.

One can restricts to bounded observables $p: X \to \mathbb{R}$ for which there is a positive bound $B \in \mathbb{R}$ such that $-B \leq p(x) \leq B$ for all $x \in X$. The collection of such bounded observables forms an order unit space [1, 51, 75, 79].

On $\text{Obs}(X) = \mathbb{R}^X$ there is also a commutative monoid structure $(1, \&)$, where $\&$ is pointwise multiplication: $(p \& q)(x) = p(x) \cdot q(x)$. We prefer to write this operation as logical conjunction because that is what it is when restricted to predicates. Besides, having yet another operation that is written as dot $\cdot$ might be confusing. We will occasionally write $p^n$ for $p \& \cdots \& p$ ($n$ times).

### 2.2.2 Factors

We recall that a factor is a non-negative observables and that we write $\text{Fact}(X) = (\mathbb{R}_{\geq 0})^X$ for the set of factors on a set $X$. Factors occur in probability theory for instance in the so-called sum-product algorithms for computing marginals and posterior distributions in graphical models. These matters are postponed; here we concentrate on the mathematical properties of factors.

The set of $\text{Fact}(X)$ looks like a vector space, expect that there are no negatives. Using the notation introduced in the previous subsection, we have:

$$\text{Fact}(X) = \{ p \in \text{Obs}(X) \mid p \geq 0 \}.$$ 

Factors can be added pointwise, with unit element $0 \in \text{Fact}(X)$, like random variables. But a factor $p \in \text{Fact}(X)$ cannot be re-scaled with an arbitrary real number, but only with a non-negative number $s \in \mathbb{R}_{\geq 0}$, giving $s \cdot p \in \text{Fact}(X)$. The structures are often called cones.

The monoid $(1, \&)$ on $\text{Obs}(X)$ restricts to $\text{Fact}(X)$, since $1 \geq 0$ and if $p, q \geq 0$ then also $p \& q \geq 0$.

### 2.2.3 Predicates

We first note that the set $\text{Pred}(X) = [0, 1]^X$ of predicates on a set $X$ contains the falsity $0$ and truth $1$ random variables, which are always $0$ (resp. $1$). But there are some noteworthy differences between predicates on the one hand and observables and factors on the other hand.

- Predicates are not closed under addition, since the sum of two numbers in $[0, 1]$ may ly outside $[0, 1]$. Thus, addition of predicates is a partial operation, and is then written as $p \oplus q$. Thus: $p \oplus q$ is defined if $p(x) + q(x) \leq 1$ for all $x \in X$, and in that case $(p \oplus q)(x) = p(x) + q(x)$. 

73
This operation \( \odot \) is commutative and associative in a suitably partial sense. Moreover, it has 0 has unit element: \( p \odot 0 = p = 0 \odot p \). This is structure \((\text{Pred}(X), 0, \odot)\) is called a partial commutative monoid.

- There is a ‘negation’ of predicates, written as \( p^\perp \), and called orthosupplement. It is defined as \( p^\perp = 1 - p \), that is, as \( p^\perp(x) = 1 - p(x) \). Then: \( p \odot p^\perp = 1 \) and \( p^{\perp \perp} = p \).

- Predicates are closed under scalar multiplication \( s \cdot p \), but only if one restricts the scalar \( s \) to be in the unit interval \([0, 1]\). It interacts nicely with partial addition \( \odot \), in the sense that \( s \cdot (p \odot q) = (s \cdot p) \odot (s \cdot q) \).

The combination of these points means that the set \( \text{Pred}(X) \) carries the structure of an effect module \([41]\), see also \([28]\). These structures arose in mathematical physics \([30]\) in order to axiomatise the structure of quantum predicates on Hilbert spaces.

The effect module \( \text{Pred}(X) \) also carries a commutative monoid structure for conjunction, namely \((1, \&)\). Indeed, when \( p, q \in \text{Pred}(X) \), then also \( p \& q \in \text{Pred}(X) \). We have \( p \& 0 = 0 \) and \( p \& (q_1 \odot q_2) = (p \& q_1) \odot (p \& q_2) \).

Since these effect structures are not so familiar, we include more formal descriptions.

**Definition 2.2.1.** 1 A partial commutative monoid (PCM) consists of a set \( M \) with a zero element \( 0 \in M \) and a partial binary operation \( \odot: M \times M \rightarrow M \) satisfying the three requirements below. They involve the notation \( x \perp y \) for:

1. Commutativity: \( x \perp y \) implies \( y \perp x \) and \( x \odot y = y \odot x \);
2. Associativity: \( y \perp z \) and \( x \perp (y \odot z) \) implies \( x \perp y \) and \( (x \odot y) \perp z \) and also \( x \odot (y \odot z) = (x \odot y) \odot z \);
3. Zero: \( 0 \perp x \) and \( 0 \odot x = x \);

2 An effect algebra is a PCM \((E, 0, \odot)\) with an orthosupplement. The latter is a total unary ‘negation’ operation \((\cdot)^\perp: E \rightarrow E \) satisfying:

1. \( x^\perp \in E \) is the unique element in \( E \) with \( x \odot x^\perp = 1 \), where \( 1 = 0^\perp \);
2. \( x \perp 1 \Rightarrow x = 0 \).

A homomorphism \( E \rightarrow D \) of effect algebras is given by a function \( f: E \rightarrow D \) between the underlying sets satisfying \( f(1) = 1 \), and if \( x \perp x' \in E \) then both \( f(x) \perp f(x') \) in \( D \) and \( f(x \odot x') = f(x) \odot f(x') \). Effect algebras and their homomorphisms form a category, denoted by \( \text{EA} \).

4 An effect module is an effect algebra \( E \) with a scalar multiplication \( s \cdot x \), for \( s \in [0, 1] \) and \( x \in E \) forming an action:

\[
1 \cdot x = x \quad (r \cdot s) \cdot x = r \cdot (s \cdot x),
\]
and preserving sums (that exist) in both arguments:

\[
\begin{align*}
0 \cdot x &= 0 \\
(s \cdot 0) &= 0 \\
(r + s) \cdot x &= r \cdot x \oplus s \cdot x \\
(s \cdot (x \otimes y)) &= s \cdot x \otimes s \cdot y.
\end{align*}
\]

We write \( \text{EMod} \) for the category of effect modules, where morphisms are maps of effect algebras that preserve scalar multiplication (i.e. are `equivariant`).

### 2.2.4 Sharp predicates

The structure of the set \( \text{SPred}(X) = \{0, 1\}^X \) is most familiar to logicians: they form Boolean algebras. The join \( p \vee q \) of \( p, q \in \text{SPred}(X) \) is the point-wise join \((p \vee q)(x) = p(x) \vee q(x)\), where the latter disjunction \( \vee \) is the obvious one on \([0, 1]\). One can see these disjunctions \((0, \vee)\) as forming an additive structure with negation \((-)\). Conjunction \((1, \&\)) forms a commutative structure on top.

Formally one can say that \( \text{SPred}(X) \) also has scalar multiplication, with scalars from \([0, 1]\), in such a way that \( 0 \cdot p = 0 \) and \( 1 \cdot p = p \).

In summary, observables all share the same multiplicative structure \((1, \&\)) for conjunction, but the additive structures and scalar multiplications differ. The latter, however, are preserved under taking validity, and not \((1, \&\))

**Lemma 2.2.2.** Let \( \omega \in \mathcal{D}(X) \) be a distribution on a set \( X \). Operations on observables \( p, q \) on \( X \) satisfy, whenever defined,

1. \( \omega \models 0 = 0 \);
2. \( \omega \models (p + q) = (\omega \models p) + (\omega \models q) \);
3. \( \omega \models (s \cdot p) = s \cdot (\omega \models p) \).

### 2.2.5 Parallel products and weakening

Earlier we have seen parallel products \( \otimes \) of states and of channels. This \( \otimes \) can also be defined for observables, and is then often called parallel conjunction. The difference between parallel conjunction \( \otimes \) and sequential conjunction \( \& \) is that \( \otimes \) acts on observables on different sets \( X, Y \) and yields an outcome on the product set \( X \times Y \), whereas \( \& \) works for observables on the same set \( Z \), and produces a conjunction observable again on \( Z \). These \( \otimes \) and \( \& \) are inter-definable, via transformation \( \ll \) of observables, see Section 2.4 in particular Exercise 2.4.5.
Chapter 2. Predicates and Observables

Definition 2.2.3. 1 Let \( p \) be an observable on a set \( X \), and \( q \) on \( Y \). Then we define a new observable \( p \otimes q \) on \( X \times Y \) by:

\[
(p \otimes q)(x, y) := p(x) \cdot q(x).
\]

2 Suppose we have an observable \( p \) on a set \( X \) and we like to use \( p \) on the product \( X \times Y \). This can be done by taking \( p \otimes 1 \) instead. This \( p \otimes 1 \) is called a \textit{weakening} of \( p \). It satisfies \( (p \otimes 1)(x, y) = p(x) \).

More generally, for a projection \( \pi_i : X_1 \times \cdots \times X_n \to X_i \) and an observable \( p \) on \( X_i \), we can take as weakening of \( p \) the predicate on \( X_1 \times \cdots \times X_n \) given by:

\[
1 \otimes \cdots \otimes 1 \otimes p \otimes 1 \otimes \cdots \otimes 1,
\]

where the \( 1 \) in the latter marginalisation mask is at position \( i \).

Weakening is a structural operation in logic which makes it possible to use a predicate \( p(x) \) depending on a single variable \( x \) in a larger context where one has for instance two variables \( x, y \) by ignoring the additional variable \( y \). Weakening is usually not an explicit operation, except in settings like linear logic where one has to be careful about the use of resources. Here, we need weakening as an explicit operation in order to avoid type mismatches between observables and underlying sets.

Recall that marginalisation of states is an operation that moves a state to a smaller underlying set by projecting away. Weakening can be seen as a dual operation, moving an observable to a larger context. There is a close relationship between marginalisation and weakening via validity: for a state \( \omega \in \mathcal{D}(X_1 \times \cdots \times X_n) \) and an observable \( p \) on \( X_i \), we have:

\[
\omega \models 1 \otimes \cdots \otimes 1 \otimes p \otimes 1 \otimes \cdots \otimes 1 = \mathcal{D}(\pi_i)(\omega) \models p = \omega[0, \ldots, 0, 1, 0, \ldots, 0] \models p,
\]

where the \( 1 \) in the latter marginalisation mask is at position \( i \). Soon we shall see an alternative description of weakening in terms of predicate transformation. The above equation then appears as a special case of a more general result, namely of Proposition 2.4.3.

2.2.6 Functoriality

We like to conclude this section with some categorical observations. They are not immediately relevant for the sequel and may be skipped. We shall be using four (new) categories:

- **Vect**, with vector spaces (over the real numbers) as objects and linear maps as morphisms between them (preserving addition and scalar multiplication);
2.2. The structure of observables

- **Cone**, with cones as objects and also with linear maps as morphisms, but this time preserving scalar multiplication with non-negative reals only;
- **EMod**, with effect modules as objects and homomorphisms of effect modules as maps between them (see Definition 2.2.1);
- **BA**, with Boolean algebras as objects and homomorphisms of Boolean algebras (preserving finite joins and negations, and then also finite meets).

Recall from Subsection 1.10.1 that we write \( \mathbb{C}^{\text{op}} \) for the opposite of category \( \mathbb{C} \), with arrows reversed. This opposite is needed in the following result.

**Proposition 2.2.4.** Taking particular observables on a set is functorial: there are functors:

1. \( \text{Obs} : \text{Sets} \to \text{Vect}^{\text{op}} \);
2. \( \text{Fact} : \text{Sets} \to \text{Cone}^{\text{op}} \);
3. \( \text{Pred} : \text{Sets} \to \text{EMod}^{\text{op}} \);
4. \( \text{SPred} : \text{Sets} \to \text{BA}^{\text{op}} \).

On maps \( f : X \to Y \) in \( \text{Sets} \) these functors are all defined by the ‘pre-compose with \( f \)’ operation \( q \mapsto q \circ f \). They thus reverse the direction of morphisms, which necessitates the use of opposite categories \( (\cdots)^{\text{op}} \).

The above functors all preserve the partial order on observables and also the commutative monoid structure \( (1, \& ) \), since they are defined point-wise.

**Proof.** We consider the first instance of observables in some detail. The other cases are similar. For a set \( X \) we have seen that \( \text{Obs}(X) = \mathbb{R}^X \) is a vector space, and thus an object of the category \( \text{Vect} \). Each function \( f : X \to Y \) in \( \text{Sets} \) gives rise to a function \( \text{Obs}(f) : \text{Obs}(Y) \to \text{Obs}(X) \) in the opposite direction. It maps an observable \( q : Y \to \mathbb{R} \) on \( Y \) to the observable \( q \circ f : X \to \mathbb{R} \) on \( X \). It is not hard to see that this function \( \text{Obs}(f) = (\_ \circ f) \) preserves the vector space structure. For instance, it preserves sums, since they are defined point-wise. We shall prove this in a precise, formal manner. First \( \text{Obs}(f)(0) \) is the function that maps \( x \in X \) to:

\[
\text{Obs}(f)(0)(x) = (0 \circ f)(x) = 0(f(x)) = 0.
\]

Hence \( \text{Obs}(f)(0) \) maps everything to 0 and is thus equal to the zero function itself: \( \text{Obs}(f)(0) = 0 \). Next, addition \( + \) is preserved since:

\[
\text{Obs}(f)(p + q) = (p + q) \circ f
= [x \mapsto (p + q)(f(x))]
= [x \mapsto p(f(x)) + q(f(x))]
= [x \mapsto \text{Obs}(f)(p)(x) + \text{Obs}(f)(q)(x)]
= \text{Obs}(f)(p) + \text{Obs}(f)(q).
\]
We leave preservation of scalar multiplication to the reader and conclude that \( \text{Obs}(f) \) is a linear function, and thus a morphism \( \text{Obs}(f) : \text{Obs}(Y) \to \text{Obs}(X) \) in \( \textbf{Vect} \). Hence \( \text{Obs}(f) \) is a morphism \( \text{Obs}(X) \to \text{Obs}(Y) \) in the opposite category \( \textbf{Vect}^{\text{op}} \). We still need to check that identity maps and composition is preserved. We do the latter. For \( f : X \to Y \) and \( g : Y \to Z \) in \( \textbf{Sets} \) we have, for \( r \in \text{Obs}(Z) \),

\[
\begin{align*}
\text{Obs}(g \circ f)(r) &= r \circ (g \circ f) = (r \circ g) \circ f \\
&= \text{Obs}(g)(r) \circ f \\
&= \text{Obs}(f)(\text{Obs}(g)(r)) \\
&= (\text{Obs}(f) \circ \text{Obs}(g))(r) \\
&= (\text{Obs}(g) \circ \text{op} \text{Obs}(f))(r).
\end{align*}
\]

This yields \( \text{Obs}(g \circ f) = \text{Obs}(g) \circ \text{op} \text{Obs}(f) \), so that we get a functor of the form \( \text{Obs} : \textbf{Sets} \to \textbf{Vect}^{\text{op}} \).

Notice that saying that we have a functor like \( \text{Pred} : \textbf{Sets} \to \textbf{EMod}^{\text{op}} \) contains remarkably much information, about the mathematical structure on objects \( \text{Pred}(X) \), about preservation of this structure by maps \( \text{Pred}(f) \), and about preservation of identity maps and composition by \( \text{Pred}(−) \) on morphisms. This makes the language of category theory both powerful and efficient.

### Exercises

2.2.1 Find examples of predicates \( p, q \) on a set \( X \) and a distribution \( \omega \) on \( X \) such that \( \omega \models p & q \) and \( (\omega \models p) \cdot (\omega \models q) \) are different.

2.2.2 1 Check that (sharp) indicator predicates \( 1_E : X \to [0, 1] \), for subsets \( E \subseteq X \), satisfy:

- \( 1_{E \cap D} = 1_E \& 1_D \);
- \( 1_{E \cup D} = 1_E \oplus 1_D \), if \( E, D \) are disjoint;
- \( (1_E)^\perp = 1_{\neg E} \), where \( \neg E = \{ x \in X \mid x \notin E \} \) is the complement of \( E \).

Formally, the function \( 1_{−} : \mathcal{P}(X) \to \text{Pred}(X) = [0, 1]^X \) is a homomorphism of effect algebras, see \([28, 41]\) for details.

2 Now consider subsets \( E \subseteq X \) and \( D \subseteq Y \) of different sets \( X, Y \) together with their product subset \( E \times D \subseteq X \times Y \). Show that \( 1_{E \times D} = 1_E \otimes 1_D \), with as special case \( 1_{(x,y)} = 1_x \otimes 1_y \).

2.2.3 Show that \( 1 \otimes 1 = 1 \).

2.2.4 1 Let \( p_i, q_i \) be observables on \( X_i \). Prove that:

\[
(p_1 \otimes \cdots \otimes p_n) \& (q_1 \otimes \cdots \otimes q_n) = (p_1 \& q_1) \otimes \cdots \otimes (p_n \& q_n).
\]
2.3 Conditioning

What we call conditioning is sometimes called update, or belief update. It is the incorporation of evidence into a distribution, where the evidence is given by a predicate (or more generally by a factor). This section describes the definition and basic results, including Bayes’ theorem. The relevance of conditioning in probabilistic reasoning will be demonstrated in many examples later on in this chapter.

Definition 2.3.1. Let \( \omega \in \mathcal{D}(X) \) be a distribution on a sample space \( X \) and \( p \in \text{Fact}(X) = (\mathbb{R}_{\geq 0})^X \) be a factor, on the same space \( X \).

1. If the validity \( \omega \models p \) is non-zero, we define a new distribution \( \omega|_p \in \mathcal{D}(X) \) as:

\[
\omega|_p(x) := \frac{\omega(x) \cdot p(x)}{\omega \models p}, \quad i.e., \quad \omega|_p = \sum_x \frac{\omega(x) \cdot p(x)}{\omega \models p} [x].
\]

This distribution \( \omega|_p \) may be pronounced as: \( \omega \), given \( p \). It is the result of conditioning/updating \( \omega \) with \( p \) as evidence.

2. The conditional expectation or conditional validity of an observable \( q \) on \( X \), given \( p \) and \( \omega \), is the validity:

\[
\omega|_p \models q.
\]
For a channel \( c: X \rightarrow Y \) and a factor \( q \) on \( Y \) we define the updated channel \( c_q: X \rightarrow Y \) via pointwise updating:

\[
   c_q(x) := c(x) q
\]

In writing \( c_q \) we assume that validity \( c(x) \models q = (c \ll q)(x) \) is non-zero, for each \( x \in X \).

Often, the state \( \omega \) before updating is called the prior or the a priori state, whereas the state \( \omega_q \) is called the posterior or the a posteriori state.

The conditioning \( c_q \) of a channel in point (3) is in fact a generalisation of the conditioning of a state \( \omega_p \) in point (1), since the state \( \omega \in D(X) \) can be seen as a channel \( \omega: 1 \rightarrow X \) with a one-element set \( 1 = \{0\} \) as domain. We shall see the usefulness of conditioning of channels in Subsection 2.5.1. For now we only notice that updating the identity channel has no effect: \( \text{unit}_q = \text{unit} \).

The conditional probability notation \( P(E \mid D) \) for events \( E, D \subseteq X \) corresponds to the validity \( \omega_{1D} \models 1_E \) of the sharp predicate \( 1_E \) in the state \( \omega \) updated with the sharp predicate \( 1_D \). Indeed,

\[
   \begin{align*}
   \omega_{1D} \models 1_E &= \sum_{x \in E} \omega_{1D}(x) \cdot 1_E(x) \\
   &= \sum_{x \in E} \frac{\omega(x) \cdot 1_D(x)}{\omega \models 1_D} \\
   &= \frac{\sum_{x \in E \cap D} \omega(x)}{\sum_{x \in D} \omega(x)} = \frac{P(E \cap D)}{P(D)} = P(D \mid E).
   \end{align*}
\]

The formulation \( \omega_p \) of conditioning that is used above involves no restriction to sharp predicates, but works much more generally for fuzzy predicates/factors \( p \). This is sometimes called updating with soft or uncertain evidence \[11, 20, 49\]. It is what we use right from the start.

\textbf{Example 2.3.2.} Let’s take the numbers of a dice as sample space: \( \text{pips} = \{1, 2, 3, 4, 5, 6\} \), with a fair/uniform dice distribution \( \text{dice} = \nu_{\text{pips}} = \frac{1}{6}\langle 1 \rangle + \frac{1}{6}\langle 2 \rangle + \frac{1}{6}\langle 3 \rangle + \frac{1}{6}\langle 4 \rangle + \frac{1}{6}\langle 5 \rangle + \frac{1}{6}\langle 6 \rangle \). We consider the predicate \( \text{evenish} \in \text{Pred}(\text{pips}) = [0, 1]^{\text{pips}} \) expressing that we are fairly certain of the outcome being even:

\[
   \begin{align*}
   \text{evenish}(1) &= \frac{1}{2} \\
   \text{evenish}(2) &= \frac{2}{3} \\
   \text{evenish}(3) &= \frac{1}{10} \\
   \text{evenish}(4) &= \frac{3}{5} \\
   \text{evenish}(5) &= \frac{1}{10} \\
   \text{evenish}(6) &= \frac{1}{5}
   \end{align*}
\]
2.3. Conditioning

We first compute the validity of `evenish` for our fair dice:

\[
\text{dice} \models \text{evenish} = \sum_x \text{dice}(x) \cdot \text{evenish}(x) \\
= \frac{1}{6} \cdot \frac{1}{2} + \frac{1}{6} \cdot \frac{1}{2} + \frac{1}{6} \cdot \frac{1}{2} + \frac{1}{6} \cdot \frac{1}{2} + \frac{1}{6} \cdot \frac{1}{2} + \frac{1}{6} \cdot \frac{1}{2} \\
= \frac{2 \times 9 + 1 + 9 + 1 + 8}{60} = \frac{1}{2}.
\]

If we take `evenish` as evidence, we can update our state and get:

\[
\begin{align*}
\text{dice}_{\text{evenish}} &= \sum_x \frac{\text{dice}(x) \cdot \text{evenish}(x)}{\text{dice} \models \text{evenish}} \mid x \\
&= \frac{1}{4} \{1 \} + \frac{1}{2} \{1 \} + \frac{1}{2} \{2 \} + \frac{1}{2} \{3 \} \\
&\quad + \frac{1}{2} \{4 \} + \frac{1}{2} \{5 \} + \frac{1}{2} \{6 \} \\
&= \frac{1}{15}(1) + \frac{3}{10}(2) + \frac{6}{10}(3) + \frac{6}{10}(4) + \frac{1}{30}(5) + \frac{1}{15}(6).
\end{align*}
\]

As expected, the probabilities of the even pips is now, in the posterior, higher than the odd ones.

2 The following alarm example is due to Pearl [80]. It involves an ‘alarm’ set \(A = \{a, a^+\}\) and a ‘burglary’ set \(B = \{b, b^+\}\), with the following a priori joint distribution \(\omega \in \mathcal{D}(A \times B)\).

\[
0.000095|a, b^+| + 0.009999|a, b^+| + 0.000005|a^+, b^+| + 0.989901|a^+, b^+|.
\]

The \textit{a priori} burglary distribution is the second marginal:

\[
\omega(0, 1) = \frac{0.0001}{b^+} + \frac{0.9999}{b^+}.
\]

Someone reports that the alarm went off, but with only 80% certainty because of deafness. This can be described as a predicate \(p: A \rightarrow [0, 1]\) with \(p(a) = 0.8\) and \(p(a^+) = 0.2\). We see that there is a mismatch between the \(p\) on \(A\) and \(\omega\) on \(A \times B\). This can be solved via weakening \(p\) to \(p \otimes 1\), so that it becomes a predicate on \(A \times B\). Then we can take the update:

\[
\omega_{p \otimes 1}
\]

In order to do so we first compute the validity:

\[
\omega \models p \otimes 1 = \sum_{x,y} \omega(x, y) \cdot p(x) = 0.206.
\]

We can then compute the updated joint state as:

\[
\begin{align*}
\omega_{p \otimes 1} &= \sum_{x,y} \frac{\omega(x,y) \cdot p(x)}{\omega \models p} \mid x,y \\
&= 0.0003688|a, b^+| + 0.03882|a, b^+| \\
&\quad + 0.000004853|a^+, b^+| + 0.9608|a^+, b^+|.
\end{align*}
\]
The resulting posterior burglary distribution — with the alarm evidence taken into account — is obtained by taking the second marginal of the updated distribution:

$$(\omega_{(p \& 1)}|0, 1] = 0.0004|b) + 0.9996|b^c).$$

We see that the burglary probability is four times higher in the posterior than in the prior. What happens is noteworthy: evidence about one component $A$ changes the probabilities in another component $B$. This ‘crossover influence’ (in the terminology of [54]) or ‘crossover inference’ happens precisely because the joint distribution $\omega$ is entwined, so that the different parts can influence each other. We shall return to this theme repeatedly, see for instance in Corollary 2.5.9 below.

One of the main results for conditioning is Bayes’ theorem. We present it here for factors, and not just for sharp predicates (events).

**Proposition 2.3.3.** Let $\omega$ be distribution on a sample space $X$, and let $p, q$ be factors on $X$.

1. The product rule holds:

$$\omega|p = q = \frac{\omega = p \& q}{\omega = p}. \quad (\omega|q|p \equiv p)$$

2. Bayes’ rule holds:

$$\omega|p = q = \frac{(\omega|q|p \equiv p) \cdot (\omega|q)}{\omega = p}. \quad (\omega|q|p \equiv p)$$

This result carefully distinguishes a product rule, in point 1, and Bayes’ rule, in point 2. This distinction is not always made, and the product rule is sometimes also called Bayes’ rule.

**Proof.** 1 We straightforwardly compute:

$$\omega|p = q = \sum_x \omega|p(x) \cdot q(x) = \sum_x \frac{\omega(x) \cdot p(x)}{\omega = p} \cdot q(x)$$

$$= \sum_x \frac{\omega(x) \cdot p(x) \cdot q(x)}{\omega = p}$$

$$= \sum_x \frac{\omega(x) \cdot (p \& q)(x)}{\omega = p} = \frac{\omega \equiv p \& q}{\omega = p}.$$
2.3. Conditioning

Example 2.3.4. We instantiate Proposition 2.3.3 with sharp predicates \(1_E, 1_D\) for subsets/events \(E, D \subseteq X\). Then the familiar formulations of the product/Bayes rule appear.

1. The product rule specialises to the definition of conditional probability:

\[
P(E \mid D) = \omega_{1_E} \mid 1_E = \frac{\omega \mid 1_D \& 1_E}{\omega \mid 1_D} = \frac{\omega \mid 1_{D \cap E}}{P(D)} = \frac{P(D \cap E)}{P(D)}.
\]

2. Bayes rule, in the general formulation of Proposition 2.3.3 (2) specialises to:

\[
P(E \mid D) = \omega_{1_E} \mid 1_E = \frac{\omega \mid 1_D \cdot (\omega \mid 1_D)}{\omega \mid 1_E} = \frac{P(D \mid E) \cdot P(E)}{P(D)}.
\]

We add a few more basic facts about conditioning.

Lemma 2.3.5. Let \(\omega\) be distribution on \(X\), with factors \(p, q \in \text{Fact}(X)\).

1. Conditioning with truth has no effect:

\[
\omega \mid 1 = \omega.
\]

2. Conditioning with a point predicate gives a point state: for \(a \in X\),

\[
\omega_{1_a} = 1[a).
\]

3. Iterated conditionings commute:

\[
(\omega \mid p)_{\mid q} = \omega_{\mid p & q} = (\omega_{\mid q})_{\mid p}.
\]

4. Conditioning is stable under multiplication of the factor with a scalar \(s > 0\):

\[
\omega_{s \cdot p} = \omega_{p}.
\]

5. Conditioning can be done component-wise, for product states and factors:

\[
(\sigma \otimes \tau \mid p \otimes q) = (\sigma_{\mid p}) \otimes (\tau_{\mid q}).
\]

6. Marginalisation of a conditioning with a (similarly) weakened predicate is conditioning of the marginalised state:

\[
\omega_{1_{\neg q}[0, 1]} = \omega_{[0, 1]}_{\mid q}.
\]

7. For a function \(f : X \rightarrow Y\), used as deterministic channel,

\[
(f \gg \omega)_{\mid q} = f \gg (\omega_{\mid q \circ f}).
\]

Proof. 1. Trivial since \(\omega \mid 1 = 1\).

2. Assuming \(\omega(a) \neq 0\) we get for each \(x \in X\),

\[
\omega_{1_a}(x) = \frac{\omega(x) \cdot 1_a(x)}{\omega \mid 1_a} = \frac{\omega(a) \cdot 1[a](x)}{\omega(a)} = 1[a](x).
\]
3 It suffices to prove:
\[(\omega|_p|_q)(x) = \omega|_p(x) \cdot q(x) \quad \omega|_p \models q \quad \implies \quad \omega|_p(x) \cdot p(x) \quad \omega \models p \quad \implies \quad \omega|_p \cdot q(x) \quad \omega \models p \quad \implies \quad \omega|_p \cdot q(x).
\]

by Proposition 2.3.3 (1)

4 First we have:
\[\omega \models s \cdot p = \sum_s \omega(x) \cdot (s \cdot p)(x) = \sum_s \omega(x) \cdot s \cdot p(x) = s \cdot \sum_s \omega(x) \cdot p(x) = s \cdot (\omega \models p).
\]

Next:
\[\omega|_{s \cdot p}(x) = \frac{\omega(x) \cdot (s \cdot p)(x)}{\omega \models s \cdot p} = \frac{\omega(x) \cdot s \cdot p(x)}{s \cdot (\omega \models p)} = \frac{\omega(x) \cdot p(x)}{\omega \models p} = \omega|_{p \cdot q}(x).
\]

5 For states \(\sigma \in D(X)\), \(\tau \in D(Y)\) and factors \(p\) on \(X\) and \(q\) on \(Y\) one has:
\[\left(\left((\sigma \otimes \tau) \right) \right\vert_{(p \otimes q)}(x, y) = \frac{(\sigma \otimes \tau)(x, y) \cdot (p \otimes q)(x, y)}{(\sigma \otimes \tau) \models (p \otimes q)} = \frac{\sigma(x) \cdot \tau(y) \cdot p(x) \cdot q(y)}{(\sigma \models p) \cdot (\tau \models q)} = \frac{\sigma(x) \cdot p(x) \cdot \tau(y) \cdot q(y)}{\sigma \models p} \cdot \tau \models q = (\sigma|_p)(x) \cdot (\tau|_q)(y)
\]

by Exercise 2.2.5

6 Let \(\omega \in D(X \times Y)\) and \(q\) be a factor on \(Y\); then:
\[(\omega|_{1 \otimes [0, 1]}(x, y) = \sum_x \omega|_{1 \otimes [0, 1]}(x, y) = \sum_x \omega(x, y) \cdot (1 \otimes [0, 1])(x, y) = \sum_x \frac{\omega(x, y) \cdot (1 \otimes [0, 1])(x, y)}{\omega|_{[0, 1]} \models q} = (\omega|_{[0, 1]}|_q)(y).
\]

7 For \(f: X \rightarrow Y\), \(\omega \in D(X)\) and \(q \in \text{Fact}(Y)\),
\[\left(\left((f \gg \omega) \right) \right\vert_q(y) = D(f)(\omega)|_q(y) = \frac{D(f)(\omega)(y) \cdot q(y)}{D(f)(\omega) \models q} = \frac{\omega(x) \cdot q(y)}{\omega(x) \cdot p(x)} = \sum_{x \in f^{-1}(y)} \omega(x) \cdot q(f(x)) = \sum_{x \in f^{-1}(y)} \omega(x) \cdot q(f(x)) = \sum_{x \in f^{-1}(y)} \omega|_{q \circ f}(x) = D(f)(\omega_{q \circ f})(y) = (f \gg (\omega|_{q \circ f}))(y).
\]
2.3. Conditioning

We conclude with the use of conditioning in defining a draw from a subset. It gives a refinement of the draw operation $D$ from Definition 1.6.3.

**Definition 2.3.6.** Let $U \subseteq X$ be a non-empty subset. It gives rise to a ‘draw from $U$’ mapping

$$N(X) \xrightarrow{D_U} D(X \times N(X))$$

given by:

$$D_U(\phi) := D(\phi)|_{x \neq \emptyset} = \sum_{x \in U} \frac{Flr(\phi)(x)}{\sum_{y \in U} Flr(\phi)(y)} \langle x, D_x(\phi) \rangle.$$

Notice that $D_X = D$, for the largest subset $X \subseteq X$.

**Example 2.3.7.** The so-called Monty Hall problem is a famous riddle in probability due to [91], see also e.g. [36, 97].

Suppose you’re on a game show, and you’re given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what’s behind the doors, opens another door, say No. 3, which has a goat. He then says to you, “Do you want to pick door No. 2?” Is it to your advantage to switch your choice?

We describe the three doors via a set $X = \{1, 2, 3\}$ and the situation in which to make the first choice as a multiset $\phi = 1|1\rangle + 1|2\rangle + 1|3\rangle$. One may also view $\phi$ as an urn from which to draw numbered balls. Let’s assume without loss of generality that the car is behind door 2. We shall represent this logically by using a (sharp) ‘goat’ predicate $1_G: \{1, 2, 3\} \to [0, 1]$, for $G = \{1, 3\}$.

We analyse the situation via the draw maps $D$ from Definition 1.6.3 and $D_G$ from Definition 2.3.6. This $D$ gives, after your first choice, the first distribution below, on $\{1, 2, 3\} \times \mathcal{M}(\{1, 2, 3\})$.

$$D(\phi) = \frac{1}{2}|1, 1|2\rangle + 1|3\rangle\rangle + \frac{1}{2}|2, 1|1\rangle + 1|3\rangle\rangle + \frac{1}{2}|3, 1|1\rangle + 1|2\rangle\rangle.$$

In the outer ket expressions $\langle \cdots , \cdots \rangle$ the first element is your choice, and the second element is the multiset with remaining options.

The host’s opening of a door with a goat is modeled by another draw, but now using the draw map $D_G$ from Definition 2.3.6, which ensures that the outcome is in $G = \{1, 3\}$. It yields on the three multisets obtained above:

$$D_G(1|2\rangle + 1|3\rangle) = 1|3, 1|2\rangle\rangle$$
$$D_G(1|1\rangle + 1|3\rangle) = \frac{1}{2}|1, 1|3\rangle\rangle + \frac{1}{2}|3, 1|1\rangle\rangle$$
$$D_G(1|1\rangle + 1|2\rangle) = 1|1, 1|2\rangle\rangle.$$
Your choice, followed by the host’s choice, can be described via channel composition, like in Example 1.8.4:

\[
((\text{id} \otimes D_G) \circ D)(\varphi) = \frac{1}{4}[1, 3, 1|2\rangle + \frac{1}{6}[2, 1, 1|3\rangle + \frac{1}{6}[2, 3, 1|1\rangle + \frac{1}{3}[3, 1, 1|2\rangle].
\]

Inside \(\cdot\) the first component is your choice, the second component is the host’s choice, and the third component contains the remaining option.

Inspection of this outcome shows that in two out of three cases — when you choose 1 or 3, in the first and last term — it makes sense to change your choice, because the car is behind the remaining door 2. Hence changing is better than sticking to your original choice.

We have given a formal account of the situation. More informally, the host knows where the car is, so his choice is not arbitrary. By opening a door with a goat behind it, the host is giving you information that you can exploit to improve your choice: two of your possible choices are wrong, but in those two out three cases the host gives you information how to correct your choice.

**Exercises**

2.3.1 In the setting of Example 2.3.2 (1) define a new predicate \(\text{oddish} = \text{evenish}^+ = 1 - \text{evenish}\).

1. Compute \(\text{dice}|_{\text{oddish}}\).
2. Prove the equation below, involving a convex sum of states on the left-hand-side (see before Subsection 1.5.1).

\[
(\text{dice} \models \text{evenish}) \cdot \omega|_{\text{evenish}} + (\text{dice} \models \text{oddish}) \cdot \omega|_{\text{oddish}} = \text{dice}.
\]

2.3.2 Let \(p_1, \ldots, p_n\) be an \(n\)-tuple of predicates on \(X\) with \(p_1 \otimes \cdots \otimes p_n = 1\). Such a tuple is also called an \(n\)-test or simply a test. Let \(\omega \in D(X)\) and \(q \in \text{Fact}(X)\).

1. Check that \(1 = \sum_i \omega \models p_i\).
2. Prove what is called the law of total probability:

\[
\omega = \sum_i (\omega \models p_i) \cdot \omega|_{p_i}.
\]

What happens to the expression on the right-hand-side if one of the \(p_i\) has validity zero? Check that this equation generalises Exercise 2.3.1.

(The expression on the right-hand-side in (2.5) is used to turn a test into a ‘denotation’ function \(D(X) \rightarrow D(D(X))\) in [71][72], namely as \(\omega \mapsto \sum_i (\omega \models p_i) |_{\omega|_{p_i}}\). This process is described more abstractly in terms of ‘hyper normalisation’ in [43].)
2.3. Conditioning

3. Show that:
\[ \omega \models q = \sum_i \omega \models q \& p_i. \]

4. Prove now:
\[ \omega\|_q \models p_i = \frac{\omega \models q \& p_i}{\sum_j \omega \models q \& p_j}. \]

2.3.3 Check that an \( n \)-test on a set \( X \), as described in the previous exercise, can be identified with a channel \( X \to n \).

Check that this generalises Exercise 2.4.8 where a single predicate \( p \) on \( X \) can be identified with a channel \( X \to 2 \), by considering \( p \) as a \( 2 \)-test \( p, p^\perp \).

2.3.4 Let \( \omega \in \mathcal{D}(X) \) with \( x \in \text{supp}(\omega) \). Prove that:
\[ \omega \models 1_x = \omega(x) \quad \text{and} \quad \omega\|_1 = 1|_x \text{ if } \omega(x) \neq 0. \]

2.3.5 Let \( c : X \to Y \) be a channel, with state \( \omega \in \mathcal{D}(X \times Z) \).

1. Prove that for a factor \( p \in \text{Fact}(Z) \),
\[ (c \otimes \text{id}) \gg \omega\|_{1 \otimes p} = (c \otimes \text{id}) \gg (\omega\|_{1 \otimes p}). \]

2. Show also that for \( q \in \text{Fact}(Y \times Z) \),
\[ ((c \otimes \text{id}) \gg \omega\|_q)[0,1] = (\omega\|_{c(\text{id}) = q})[0,1]. \]

2.3.6 This exercise will demonstrate that conditioning may both create and remove entwinedness.

1. Write yes = \( 1_1 : 2 \to [0,1] \), where \( 2 = \{0,1\} \), and no = yes\( ^\perp = 1_0 \).

Prove that the following conditioning of a non-entwined state,
\[ \tau := (\text{flip} \otimes \text{flip})|(\text{yes} \otimes \text{yes}) \cup (\text{no} \otimes \text{no}) \]
is entwined.

2. Consider the state \( \omega \in \mathcal{D}(2 \times 2 \times 2) \) given by:
\[ \omega = \frac{1}{16}|111\rangle + \frac{1}{16}|110\rangle + \frac{1}{16}|101\rangle + \frac{1}{16}|100\rangle \]
\[ + \frac{1}{16}|011\rangle + \frac{1}{16}|010\rangle + \frac{1}{16}|001\rangle + \frac{1}{16}|000\rangle \]

Prove that \( \omega \)'s first and third component are entwined:
\[ \omega[1,0,1] \neq \omega[1,0,0] \otimes \omega[0,0,1]. \]

3. Now let \( \rho \) be the following conditioning of \( \omega \):
\[ \rho := \omega\|_{1 \otimes \text{yes} \otimes 1}. \]

Prove that \( \rho \)'s first and third component are non-entwined.
The phenomenon that entwined states become non-entwined via conditioning is called screening-off, whereas the opposite, non-entwined states becoming entwined via conditioning, is called explaining away.

2.3.7 Show that for $\omega \in D(X)$ and $p_1, p_2 \in \text{Fact}(X)$ one has:

$$(\Delta \gg \omega)\big|_{p_1 \otimes p_2} = \Delta \gg (\omega|_{p_1 \otimes p_2}).$$

Note that this is a consequence of Lemma 2.3.5 (7).

2.3.8 We have mentioned (right after Definition 2.3.1) that updating of the identity channel has no effect. Prove more generally that for an ordinary function $f : X \to Y$, updating the associated deterministic channel $\langle f \rangle : X \to Y$ has no effect:

$$\langle f \rangle|_q = \langle f \rangle.$$

2.3.9 Let $c : Z \to X$ and $d : Z \to Y$ be two channels with a common domain $Z$, and with factors $p \in \text{Fact}(X)$ and $q \in \text{Fact}(Y)$ on their codomains. Prove that the update of a tuple channel is the tuple of the updates:

$$\langle c, d \rangle|_{p \otimes q} = \langle c|_p, d|_q \rangle.$$

Prove also that for $e : U \to X$ and $f : V \to Y$,

$$\langle e \otimes f \rangle|_{p \otimes q} = (e|_p) \otimes (f|_q).$$

2.4 Transformation of observables

One of the basic operations that we have seen so far is state transformation $\gg$. It is used to transform a state $\omega$ on the domain $X$ of a channel $c : X \to Y$ into a state $c \gg \omega$ on the codomain $Y$ of the channel. This section introduces transformation of observables $\ll$. It works in the opposite direction of a channel: an observable $q$ on the codomain $Y$ of the channel is transformed into an observable $c \ll q$ on the domain $X$ of the channel. Thus, state transformation works forwardly, in the direction of the channel, whereas observable transformation $\ll$ works backwardly, against the direction of the channel. This operation $\ll$ is often applied only to predicates — and then called predicate transformation — but here we apply it more generally to observables.

This section introduces observable transformation $\ll$ and lists its key mathematical properties. In subsequent sections it will be used for probabilistic reasoning.
2.4. Transformation of observables

Definition 2.4.1. Let \( c : X \to Y \) be a channel. An observable \( q \in \text{Obs}(Y) \) is transformed into \( c \ll q \in \text{Obs}(X) \) via the definition:

\[(c \ll q)(x) \coloneqq c(x) \equiv q = \sum_y c(x)(y) \cdot q(y).\]

There is a whole series of basic facts about \( \ll \).

Lemma 2.4.2. 1 The operation \( c \ll (-) : \text{Obs}(Y) \to \text{Obs}(X) \) of transforming random variables along a channel \( c : X \to Y \) restricts to factors \( c \ll (-) : \text{Fact}(Y) \to \text{Fact}(X) \), and also to fuzzy predicates \( c \ll (-) : \text{Pred}(Y) \to \text{Pred}(X) \), but not to sharp predicates.

2 Observable transformation \( c \ll (-) \) is linear: it preserves sums \((0,+)\) of observables and scalar multiplication of observables.

3 Observable transformation preserves truth \( 1 \), but not conjunction \&.

4 Observable transformation preserves the (point-wise) order on observables:
\[q_1 \leq q_2 \text{ implies } (c \ll q_1) \leq (c \ll q_2).\]

5 Transformation along the unit channel is the identity: unit \( \ll q = q \).

6 Transformation along a composite channel is successive transformation:
\[(d \circ c) \ll q = c \ll (d \ll q).\]

7 Transformation along a trivial, deterministic channel is pre-composition:
\[f \ll q = q \circ f.\]

Proof. 1 If \( q \in \text{Fact}(Y) \) then \( q(y) \geq 0 \) for all \( y \). But then also \( (c \ll q)(x) = \sum_y c(x)(y) \cdot q(y) \geq 0 \), so that \( c \ll q \in \text{Fact}(X) \). If in addition \( q \in \text{Pred}(Y) \), so that \( q(y) \leq 1 \) for all \( y \), then also \( (c \ll q)(x) = \sum_y c(x)(y) \cdot q(y) \leq \sum_y c(x)(y) = 1 \), since \( c(x) \in \mathcal{D}(Y) \), so that \( c \ll q \in \text{Pred}(X) \).

The fact that a transformation \( c \ll p \) of a sharp predicate \( p \) need not be sharp is demonstrated in Exercise 2.4.2.

2 Easy.

3 \((c \ll 1)(x) = \sum_y c(x)(y) \cdot 1(y) = \sum_y c(x)(y) \cdot 1 = 1 \). The fact that \& is not preserved follows from Exercise 2.4.1.

4 Easy.

5 Recall that unit \( \text{unit}(x) = 1|x \), so that \((\text{unit} \ll q)(x) = \sum_y \text{unit}(x)(y) \cdot q(y) = q(x) \).

6 For \( c : X \to Y \) and \( d : Y \to Z \) and \( q \in \text{Obs}(Z) \) we have:

\[
((d \circ c) \ll q)(x) = \sum_z (d \circ c)(x)(z) \cdot q(z)
= \sum_z (\sum_y c(x)(y) \cdot d(y)(z)) \cdot q(z)
= \sum_y c(x)(y) \cdot (\sum_z d(y)(z) \cdot q(z))
= \sum_y c(x)(y) \cdot (d \ll q)(y)
= (c \ll (d \ll q))(x).
\]
7 For a function \( f : X \rightarrow Y \),
\[
(f \ll q)(x) = \sum_y \text{unit}(f(x))(y) \cdot q(y) = q(f(x)) = (q \circ f)(x).
\]

There is the following fundamental relationship between transformations \( \gg \), \( \ll \) and validity \( \models \).

**Proposition 2.4.3.** Let \( c : X \rightarrow Y \) be a channel with a state \( \omega \in D(X) \) on its domain and an observable \( q \in \text{Obs}(Y) \) on its codomain. Then:
\[
c \gg \omega \models q = \omega \models c \ll q.
\tag{2.6}
\]

**Proof.** The result follows simply by unpacking the relevant definitions:
\[
c \gg \omega \models q = \sum_y (c \gg \omega)(y) \cdot q(y) = \sum_y \left( \sum_x c(x)(y) \cdot \omega(x) \right) \cdot q(y)
= \sum_x \omega(x) \cdot \left( \sum_y c(x)(y) \cdot q(y) \right)
= \sum_x \omega(x) \cdot (c \ll q)(x)
= \omega \models c \ll q.
\]

We have already seen several instances of this basic result.

- Earlier we mentioned that marginalisation (of states) and weakening (of observables) are dual to each other, see Equation (2.4). We can now see this as an instance of (2.6), using a projection \( \pi_i : X_1 \times \cdots \times X_n \rightarrow X_i \) as (trivial) channel, in:
\[
\pi_i \gg \omega \models p = \omega \models \pi_i \ll p
\]

On the left-hand-side the \( i \)-th marginal \( \pi_i \gg \omega \models p \) is used, whereas on the right-hand-side the observable \( p \in \text{Obs}(X_i) \) is weakened to \( \pi_i \ll p \) on the product set \( X_1 \times \cdots \times X_n \).

- The first equation in Exercise 2.1.3 is also an instance of (2.6), namely for a trivial channel \( f : X \rightarrow Y \) given by a function \( f : X \rightarrow Y \), as in:
\[
f \gg \omega \models p = \omega \models f \ll p = \omega \models q \circ f \quad \text{by Lemma 2.4.2} \tag{7}.
\]

**Remark 2.4.4.** In a programming context, where a channel \( c : X \rightarrow Y \) is seen as a program taking inputs from \( X \) to outputs in \( Y \), one may call \( c \ll q \) the **weakest precondition** of \( q \), commonly written as \( \text{wp}(c, q) \), see e.g. [26] [62] [70].

We briefly explain this view.

A precondition of \( q \), w.r.t. channel \( c : X \rightarrow Y \), may be defined as an observable \( p \) on the channel’s domain \( X \) for which:
\[
\omega \models p \leq c \gg \omega \models q, \quad \text{for all states} \ \omega.
\]
Proposition 2.4.3 tells that $c \ll q$ is then a precondition of $q$. It is also the weakest, since if $p$ is a precondition of $q$, as described above, then in particular:

$$p(x) = \text{unit}(x) \models p \leq c \gg \text{unit}(x) \models q = \text{unit}(x) \models c \ll q = (c \ll q)(x).$$

As a result, $p \leq c \ll q$.

Lemma 2.4.2(6) expresses a familiar compositionality property in the theory of weakest preconditions:

$$wp(d \circ c, q) = (d \circ c) \ll q = c \ll (d \ll q) = wp(c, wp(d, q)).$$

We close this section with two topics that dig deeper into the nature of transformations. First, we relate transformation of states and observables in terms of matrix operations. Then we look closer at the categorical aspects of transformation of observables.

Remark 2.4.5. Let $c$ be a channel with finite sets as domain and codomain. For convenience we write these as $n = \{0, \ldots, n-1\}$ and $m = \{0, \ldots, m-1\}$, so that the channel $c$ has type $n \to m$. For each $i \in n$ we have that $c(i) \in D(m)$ is given by an $m$-tuple of numbers in $[0, 1]$ that add up to one. Thus we can associate an $m \times n$ matrix $M_c$ with the channel $c$, namely:

$$M_c = \begin{pmatrix}
    c(1)(1) & \cdots & c(n-1)(1) \\
    \vdots & \ddots & \vdots \\
    c(1)(m-1) & \cdots & c(n-1)(m-1)
\end{pmatrix}.$$  

By construction, the columns of this matrix add up to one. Such matrices are often called stochastic.

A state $\omega \in D(n)$ may be identified with a column vector $M_\omega$ of length $n$, as on the left below. It is then easy to see that the matrix $M_{c \gg \omega}$ of the transformed state, is obtained by matrix-column multiplication, as on the right:

$$M_\omega = \begin{pmatrix}
    \omega(0) \\
    \vdots \\
    \omega(n-1)
\end{pmatrix} \quad \text{so that} \quad M_{c \gg \omega} = M_c \cdot M_\omega.$$

Indeed,

$$(c \gg \omega)(j) = \sum_i c(i)(j) \cdot \omega(i) = \sum_i (M_c)_{ij} \cdot (M_\omega)_i = (M_c \cdot M_\omega)_j.$$

An observable $q : m \to \mathbb{R}$ on $m$ can be identified with a row vector $M_q = \begin{pmatrix}
    q_0 \\
    \vdots \\
    q_{m-1}
\end{pmatrix}$. 

...
Transformation \( c \ll q \) then corresponds to row-matrix multiplication:

\[
M_{c \ll q} = M_q \cdot M_c.
\]

Again, this is checked easily:

\[
(c \ll q)(i) = \sum_j q(j) \cdot c(i)(j) = \sum_j (M_q)_j \cdot (M_c)_{j,i} = (M_q \cdot M_c)_{j,i}.
\]

We close this section by making the functoriality of observable transformation \( \ll \) explicit, in the style of Proposition 2.2.4. The latter deals with functions, but we now consider functoriality wrt. channels, using the category \( \text{Chan} = \text{Chan}(\mathcal{D}) \) of probabilistic channels. Notice that wrt. Proposition 2.2.4 the case of sharp predicates is omitted, simply because sharp predicates are not closed under predicate transformation (see Exercise 2.4.2). Also, conjunction \& is not preserved under transformation, see Exercise 2.4.1 below.

**Proposition 2.4.6.** Taking particular observables on a set is functorial

1. \( \text{Obs} : \text{Chan} \to \text{Vect}^{\text{op}}; \)
2. \( \text{Fact} : \text{Chan} \to \text{Cone}^{\text{op}}; \)
3. \( \text{Pred} : \text{Chan} \to \text{EMod}^{\text{op}}; \)

On a channel \( c : X \to Y \), all these functors are given by transformation \( c \ll (\cdot) \), acting in the opposite direction.

**Proof.** Essentially, all the ingredients are already in Lemma 2.4.2: transformation restricts appropriately (point (1)), transformation preserves identities (point (5)) and composition (point (6)), and the relevant structure (points (2) and (3)). \( \square \)

**Exercises**

2.4.1 Consider the channel \( f : \{a, b, c\} \to \{u, v\} \) from Example 1.7.2 (4), given by:

\[
f(a) = \frac{1}{2}|u\rangle + \frac{1}{2}|v\rangle \quad f(b) = 1|u\rangle \quad f(c) = \frac{3}{4}|u\rangle + \frac{1}{4}|v\rangle.
\]

Take as predicates \( p, q : \{u, v\} \to [0, 1] \),

\[
p(u) = \frac{1}{2} \quad p(v) = \frac{3}{4} \quad q(u) = \frac{1}{4} \quad q(v) = \frac{1}{6}.
\]

Compute:

- \( f \ll p \)
- \( f \ll q \)
- \( f \ll (p \otimes q) \)
2.4. Transformation of observables

- \((f \ll p) \oplus (f \ll q)\)
- \(f \ll (p \& q)\)
- \((f \ll p) \& (f \ll q)\)

This will show that predicate transformation \(\ll\) does not preserve conjunction \&.

2.4.2 Still in the context of the previous exercise, consider the sharp (point) predicate \(1_u\) on \(\{a, b, c\}\). Show that the transformed predicate \(f \ll 1_u\) on \(\{a, b, c\}\) is not sharp. This proves that sharp predicates are not closed under predicate transformation.

2.4.3 Let \(h: X \rightarrow Y\) be an ordinary function. Recall from Lemma 2.4.2 that \(h \ll q = q \circ h\), when \(h\) is considered as a deterministic channel. Show that transformation along such deterministic channels does preserve conjunctions:

\[h \ll (q_1 \& q_2) = (h \ll q_1) \& (h \ll q_2),\]

in contrast to the findings in Exercise 2.4.1 for arbitrary channels.

2.4.4 Recall that a state \(\omega \in \mathcal{D}(X)\) can be identified with a channel \(\{1\} \rightarrow X\) with a trivial domain, and also that a predicate \(p: X \rightarrow [0, 1]\) can be identified with a channel \(X \rightarrow \{1\}\), see Exercise 2.4.8. Check that under these identifications validity \(\omega \models p\) can be identified with:

- state transformation \(p \Rightarrow \omega\);
- predicate transformation \(\omega \ll p\);
- channel composition \(p \circ \omega\).

2.4.5 This exercises shows how parallel conjunction \(\otimes\) and sequential conjunction \& are inter-definable via transformation \(\ll\), using projection channels \(\pi_i\) and copy channels \(\Delta\), that is, using weakening and contraction.

1. Let observables \(p_1\) on \(X_1\) and \(p_2\) on \(X_2\) be given. Show that on \(X_1 \times X_2\),

\[p_1 \otimes p_2 = (\pi_1 \ll p_1) \& (\pi_2 \ll p_2) = (p_1 \otimes 1) \& (1 \otimes p_2).\]

The last equation occurred already in Exercise 2.2.4.

2. Let \(q_1, q_2\) be observables on the same set \(Y\). Prove that on \(Y\),

\[q_1 \& q_2 = \Delta \ll (q_1 \otimes q_2).\]

2.4.6 Prove that:

\[(c, d) \ll (p \otimes q) = (c \ll p) \& (d \ll q)\]
\[(e \otimes f) \ll (p \otimes q) = (e \ll p) \otimes (f \ll q).\]
2.4.7 Let $c : X \to Y$ be a channel, with observables $p$ on $Z$ and $q$ on $Y \times Z$.
Check that:

$$(c \otimes \text{id}) \ll ((1 \otimes p) \& q) = (1 \otimes p) \& ((c \otimes \text{id}) \ll q).$$

(Recall that $\&$ is not preserved by $\ll$, see Exercise 2.4.1.)

2.4.8 1 Check that a predicate $p : X \to [0, 1]$ can be identified with a channel $\hat{p} : X \to 2$. Describe this $\hat{p}$ explicitly in terms of $p$.
2 Define a channel $\text{orth} : 2 \to 2$ such that $\text{orth} \circ \hat{p} = \hat{p}^\perp$.
3 Define also a channel $\text{conj} : 2 \times 2 \to 2$ such that $\text{conj} \circ (\hat{p}, \hat{q}) = \hat{p} \& \hat{q}$.
4 Finally, define also a channel $\text{scal}(r) : 2 \to 2$, for $r \in [0, 1]$, so that $\text{scal}(r) \circ \hat{p} = \hat{r} \cdot p$.

2.4.9 From a categorical perspective, predicate transformation allows us to adapt the functor

$$\text{Sets} \xrightarrow{\text{Pred}} \text{EMod}^{\text{op}}$$

from Proposition 2.2.4 to a functor:

$$\text{Chan}(\mathcal{D}) \xrightarrow{\text{Pred}} \text{EMod}^{\text{op}}$$

by redefining $\text{Pred}$ on morphisms $c : X \to \mathcal{D}(Y)$ in $\text{Chan}(\mathcal{D})$ as $\text{Pred}(c) : \text{Pred}(Y) \to \text{Pred}(X)$ with:

$$\text{Pred}(c)(q) \coloneqq c \ll q.$$

1 Check in detail that $\text{Pred}(c)$ is a morphism of effect modules. Check also that channel composition and identities are preserved — and thus that we have functors indeed.
2 Show that the functor $\text{Pred}$ is faithful, in the sense that $\text{Pred}(c) = \text{Pred}(c')$ implies $c = c'$, for channels $c, c' : X \to Y$.
3 Let $Y$ be a finite set. Show that for each map $h : \text{Pred}(Y) \to \text{Pred}(X)$ in the category $\text{EMod}$ there is a unique channel $c : X \to \mathcal{D}(Y)$ with $\text{Pred}(c) = h$.

Hint: Write a predicate $p$ as finite sum $\sum_y p(y) \cdot \mathbf{1}_y$ like in Exercise 2.2.7 and use the relevant preservation property.

One says that the functor $\text{Pred} : \text{Chan}_{\text{fin}}(\mathcal{D}) \to \text{EMod}^{\text{op}}$ is full and faithful when restricted to the category $\text{Chan}_{\text{fin}}(\mathcal{D})$ with finite sets as objects. In the context of programming (logics) this property is called healthiness, see [25, 26, 70], or [37] for an abstract account.
2.5 Reasoning along channels

The combination of \( \gg \) and \( \ll \) with conditioning of states from Section 2.3 gives rise to the powerful techniques of forward and backward inference. This will be illustrated in the current section. At the end, these channel-based inference mechanisms are related to crossover inference for joint states.

The next definition captures the two basic patterns (first formulated in [53]). We shall refer to them jointly as channel-based inference, or as reasoning along channels.

**Definition 2.5.1.** Let \( \omega \in \mathcal{D}(X) \) be a state on the domain of a channel \( c : X \rightarrow Y \).

1. For a factor \( p \in \text{Fact}(X) \), we define forward inference as transformation along \( c \) of the state \( \omega \) updated with \( p \), as in:
   \[ c \gg (\omega|_p). \]
   This is also called prediction or causal reasoning.

2. For a factor \( q \in \text{Fact}(Y) \), backward inference is updating of \( \omega \) with the transformed factor:
   \[ \omega|_{(c \ll q)}. \]
   This is also called explanation or evidential reasoning.

In both cases the distribution \( \omega \) is often called the prior distribution or simply the prior. Similarly, \( c \gg (\omega|_p) \) and \( \omega|_{(c \ll q)} \) are called posterior distributions or just posteriors.

Thus with forward inference one first conditions and then performs (forward, state) transformation, whereas for backward inference one first performs (backward, factor) transformation, and then one conditions. We shall illustrate these fundamental inferences mechanisms in several examples. An important first step in these examples is to recognise the channel that is hidden in the description of the problem at hand. It is instructive to try and do this, before reading the analysis and the solution.

**Example 2.5.2.** We start with the following question from [87, Example 1.12].

Consider two urns. The first contains two white and seven black balls, and the second contains five white and six black balls. We flip a coin and then draw a ball from the first urn or the second urn depending on whether the outcome was heads or tails. What is the conditional probability that the outcome of the toss was heads given that a white ball was selected?
Our analysis involves two sample spaces \( \{H, T\} \) for the sides of the coin and \( \{W, B\} \) for the colors of the balls in the urns. The coin distribution \( \gamma \) is uniform: \( \gamma = \frac{1}{2}|H\rangle + \frac{1}{2}|T\rangle \). The above description implicitly contains a channel \( c: \{H, T\} \to \{W, B\} \), namely:

\[
c(H) = \frac{2}{9}|W\rangle + \frac{7}{9}|B\rangle \quad \text{and} \quad c(T) = \frac{5}{11}|W\rangle + \frac{6}{11}|B\rangle.
\]

As in the above quote, the first urn is associated with heads and the second one with tails. Notice that the distributions \( c(H), c(T) \) are obtained via frequentist learning, from the two urns, seen as multisets, see Section 1.6.

The evidence that we have is described in the quote after the word ‘given’. It is the singleton predicate \( 1_W \) on the set of colours \( \{W, B\} \), indicating that a white ball was selected. This evidence can be pulled back (transformed) along the channel \( c \), to a predicate \( c \ll 1_W \) on the sample space \( \{H, T\} \). It is given by:

\[
(c \ll 1_W)(H) = \sum_{x \in \{W, B\}} c(H)(x) \cdot 1_W(x) = c(H)(W) = \frac{2}{9}.
\]

Similarly we get \( (c \ll 1_W)(T) = c(T)(W) = \frac{5}{11} \).

The answer that we are interested in is obtained by updating the prior \( \gamma \) with the transformed evidence \( c \ll 1_W \), as given by \( \gamma_{c \ll 1_W} \). This is an instance of backward inference.

In order to get this answer, we first we compute the validity:

\[
\gamma \models c \ll 1_W = \gamma(H) \cdot (c \ll 1_W)(H) + \gamma(T) \cdot (c \ll 1_W)(T) = \frac{1}{2} \cdot \frac{2}{9} + \frac{1}{2} \cdot \frac{5}{11} = \frac{1}{3} + \frac{5}{22} = \frac{67}{198}.
\]

Then:

\[
\gamma_{c \ll 1_W} = \frac{1/2 \cdot 2/9}{67/198}|H\rangle + \frac{1/2 \cdot 5/11}{67/198}|T\rangle = \frac{22}{67}|H\rangle + \frac{15}{67}|T\rangle.
\]

Thus, the conditional probability of heads is \( \frac{22}{67} \). The same outcome is obtained in [87], of course, but there via an application of Bayes’ rule.

**Example 2.5.3.** Consider the following classical question from [98].

A cab was involved in a hit and run accident at night. Two cab companies, Green and Blue, operate in the city. You are given the following data:

- 85% of the cabs in the city are Green and 15% are Blue
- A witness identified the cab as Blue. The court tested the reliability of the witness under the circumstances that existed on the night of the accident, and concluded that the witness correctly identified each one of the two colors 80% of the time and failed 20% of the time.

What is the probability that the cab involved in the accident was Blue rather than Green?
2.5. Reasoning along channels

We use as colour set $C = \{G, B\}$ for Green and Blue. There is a prior ‘base rate’ distribution $\omega = \frac{12}{20} |G\rangle + \frac{3}{20} |B\rangle \in \mathcal{D}(C)$, as in the first bullet above.

The reliability information in the second bullet translates into a ‘correctness’ channel $c: \{G, B\} \to \{G, B\}$ given by:

\[
c(G) = \frac{4}{5} |G\rangle + \frac{1}{5} |B\rangle \quad c(B) = \frac{1}{5} |G\rangle + \frac{4}{5} |B\rangle.
\]

The second bullet also gives evidence of a Blue car. It translates into a point predicate $1_B$ on $\{G, B\}$. It can be used for backward inference, giving the answer to the query, as posterior:

\[
\omega_{\ll 1_B} = \frac{17}{20} |G\rangle + \frac{3}{20} |B\rangle \approx 0.5862 |G\rangle + 0.4138 |B\rangle.
\]

Thus the probability that the Blue car was actually involved in the incident is a bit more that 41%. This may seem like a relatively low probability, given that the evidence says ‘Blue taxicab’ and that observations are 80% accurate. But this low percentage is explained by the fact that are relatively few Bleu taxicabes in the first place, namely only 15%. This is in the prior, base rate distribution $\omega$. It is argued in [98] that humans find it difficult to take such base rates (or priors) into account. They call this “base rate neglect”, see also [32].

Notice that the channel $c$ is used to accommodate the uncertainty of observations: the point observation ‘Blue taxicab’ is transformed into a predicate $c \ll 1_B$. The latter is $G \mapsto 0.2$ and $B \mapsto 0.8$. Updating $\omega$ with this predicate gives the claimed outcome.

**Example 2.5.4.** Recall the Medicine-Blood Table (1.4) with data on different types of medicine via a set $M = \{0, 1, 2\}$ and blood pressure via the set $B = \{H, L\}$. From the table we can extract a channel $b: M \to B$ describing the blood pressure distribution for each medicine type. This channel is obtained by column-wise frequentist learning:

\[
b(0) = \frac{3}{8} |H\rangle + \frac{1}{8} |L\rangle \quad b(1) = \frac{7}{9} |H\rangle + \frac{2}{9} |L\rangle \quad b(2) = \frac{5}{8} |H\rangle + \frac{3}{8} |L\rangle.
\]

The prior medicine distribution $\omega = \frac{3}{20} |0\rangle + \frac{9}{20} |1\rangle + \frac{2}{5} |2\rangle$ is obtained from the totals row in the table.

The predicted state $b \gg \omega$ is $\frac{7}{10} |H\rangle + \frac{3}{10} |L\rangle$. It is the distribution that is learnt from the totals column in Table (1.4). Suppose we wish to focus on the people that take either medicine 1 or 2. We do so by conditioning, via the subset $E = \{1, 2\} \subseteq B$, with associated sharp predicate $1_E: B \to \{0, 1\}$. Then:

\[
\omega \models 1_E = \frac{9}{20} + \frac{2}{5} = \frac{17}{20} \quad \text{so} \quad \omega |_{1_E} = \frac{9}{17} |1\rangle + \frac{2}{17} |2\rangle = \frac{9}{17} |1\rangle + \frac{8}{17} |2\rangle.
\]
Chapter 2. Predicates and Observables

Forward reasoning, precisely as in Definition 2.5.1(1), gives:

\[ b \gg (\omega | \mathbf{1}_{H}) = b \gg \left( \frac{2}{17} | 1 + \frac{5}{17} | 2 \right) \]
\[ = \left( \frac{2}{17} \cdot \frac{1}{5} + \frac{8}{17} \cdot \frac{5}{9} \right) | H + \left( \frac{2}{17} \cdot \frac{1}{9} \cdot \frac{3}{5} \right) | H \]
\[ = \frac{12}{17} | H + \frac{5}{17} | L. \]

This shows the distribution of high and low blood pressure among people using medicine 1 or 2.

We turn to backward reasoning. Suppose that we have evidence \( \mathbf{1}_{H} \) on \( \{H, L\} \) of high blood pressure. What is then the associated distribution of medicine usage? It is obtained in several steps:

\[ (b \ll \mathbf{1}_{H}) (x) = \sum_{y} b(x)(y) \cdot \mathbf{1}_{H}(y) = b(x)(H) \]
\[ \omega \models b \ll \mathbf{1}_{H} = \sum_{x} \omega(x) \cdot (b \ll \mathbf{1}_{H})(x) = \sum_{x} \omega(x) \cdot b(x)(H) \]
\[ = \frac{3}{20} \cdot \frac{2}{5} + \frac{9}{20} \cdot \frac{2}{5} + \frac{2}{5} \cdot \frac{5}{8} = \frac{7}{10} \]
\[ \omega_{\ll \mathbf{1}_{H}} = \sum_{x} \omega(x) \cdot (b \ll \mathbf{1}_{H})(x) \]
\[ = \frac{3}{10} \cdot \frac{2}{3} \cdot \frac{1}{2} + \frac{9}{20} \cdot \frac{2}{5} \cdot \frac{5}{8} \approx 0.1429(0) + 0.5(1) + 0.3571(2). \]

We can also reason with ‘soft’ evidence, using the full power of fuzzy predicates. Suppose we are only 95% sure that the blood pressure is high, due to some measurement uncertainty. Then we can use as evidence the predicate \( q : \mathcal{P} \rightarrow \{0, 1\} \) given by \( q(H) = 0.95 \) and \( q(L) = 0.05 \). It yields:

\[ \omega_{\ll q} \approx 0.1434(0) + 0.4963(1) + 0.3603(2). \]

We see a slight difference wrt. the outcome with sharp evidence.

Example 2.5.5. The following question comes from [97, §6.1.3] (and is also used in [78]).

One fish is contained within the confines of an opaque fishbowl. The fish is equally likely to be a piranha or a goldfish. A sushi lover throws a piranha into the fishbowl alongside the other fish. Then, immediately, before either fish can devour the other, one of the fish is blindly removed from the fishbowl. The fish that has been removed from the bowl turns out to be a piranha. What is the probability that the fish that was originally in the bowl by itself was a piranha?

Let’s use the letters ‘p’ and ‘g’ for piranha and goldfish. We are looking at a situation with multiple fish in a bowl, where we cannot distinguish the order. Hence we describe the contents of the bowl as a (natural) muliset over \( \{p, g\} \).
2.5. Reasoning along channels

that is, as an element of \(N((p, g))\). The initial situation can then be described as a distribution \(\omega \in D(N((p, g)))\) with:

\[
\omega = \frac{1}{2} \ket{p} + \frac{1}{2} \ket{g}.
\]

Adding a piranha to the bowl involves a function \(A: N(\{p, g\}) \rightarrow N(\{p, g\})\), such that \(A(\sigma) = (\sigma(p) + 1)|p\rangle + \sigma(g)|g\rangle\). It forms a deterministic channel.

We use a piranha predicate \(P: N((p, g)) \rightarrow [0, 1]\) that gives the likelihood \(P(\sigma)\) of taking a piranha from a multiset/bowl \(\sigma\). Thus:

\[
P(\sigma) := \text{Flrn}(\sigma)(p) \frac{\sigma(p)}{\sigma(p) + \sigma(g)}.
\]

We have now collected all ingredients to answer the question via backward inference along the deterministic channel \(A\). It involves the following steps.

\[
(A \ll P)(\sigma) = P(A(\sigma)) = \frac{\sigma(p) + 1}{\sigma(p) + 1 + \sigma(g)}
\]

\[
\omega \models A \ll P = \frac{1}{2} \cdot (A \ll P)(|p\rangle) + \frac{1}{2} \cdot (A \ll P)(|g\rangle)
\]

\[
= \frac{1}{2} \cdot \frac{1 + \frac{1}{2}}{1 + \frac{1}{2} + \frac{1}{4}} + \frac{1}{2} \cdot \frac{0 + 1 + \frac{1}{4}}{1 + \frac{1}{2} + \frac{1}{4}} = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}
\]

\[
\omega_{A \ll P} = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4} |p\rangle + \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4} |g\rangle
\]

\[
= \frac{3}{8} |p\rangle + \frac{1}{8} |g\rangle.
\]

Hence the answer to the question in the beginning of this example is: \(\frac{3}{7}\).

The previous illustrations of channel-based inference are rather concrete. The next example is more abstract and involves the binomial distribution as channel.

**Example 2.5.6.** Capture and recapture is a methodology used in ecology to estimate the size of a population. So imagine we are looking at a pond and we wish to learn the number of fish. We catch twenty of them, mark them, and throw them back. Subsequently we catch another twenty, and find out that five of them are marked. What do we learn about the number of fish?

The number of fish in the pond must be at least 20. Let’s assume the maximal number is 300. We will be considering units of 10 fish. Hence the underlying sample space \(F\) together with the uniform ‘prior’ state \(\nu_F\) is:

\[
F = \{20, 30, 40, \ldots, 300\} \quad \text{with} \quad \nu_F = \sum_{x \in F} \frac{1}{59}|x\rangle.
\]

We now assume that \(K = 20\) of the fish in the pond are marked. We can then compute for each value 20, 30, 40, \ldots in the fish space \(F\) the probability of finding 5 marked fish when 20 of them are caught. In order not to complicate the calculations too much, we catch these 20 fish one by one, check if they are...
marked, and then throw them back. This means that the probability of catching a marked fish remains the same, and is described by a binomial distribution, see Example 1.5.1(2). Its parameters are $K = 20$ with probability $\frac{i}{K}$ of catching a marked fish, where $i \in F$ is the assumed total number of fish. This is incorporated in the following ‘catch’ channel $c: F \rightarrow K + 1 = \{0, 1, \ldots, K\}$.

$$c(i) := \text{binom}[K] \left( \frac{i}{K} \right) = \sum_{k=0}^{K+1} \binom{K}{k} \left( \frac{i}{K} \right)^k \left( \frac{K-i}{K} \right)^{K-k} | k \rangle.$$ 

Once this is set up, we construct a posterior state by updating the prior with the information that five marked fish have been found. The latter is expressed as point predicate $1_5 \in \text{Pred}(K + 1)$ on the codomain of the channel $c$. We can then do backward inference, as in Definition 2.5.1(2), and obtain the updated uniform distribution:

$$\nu_{F | c \ll 1_5}.$$ 

The bar chart of this posterior is in Figure 2.1; it indicates the likelihoods of the various numbers of fish in the pond. One can also compute the expected value (mean) of this posterior; it’s 116.5 fish. In case we had caught 10 marked fish out of 20, the expected number would be 47.5.

Note that taking a uniform prior corresponds to the idea that we have no idea about the number of fish in the pond. But possibly we already had a good estimate from previous years. Then we could have used such an estimate as prior distribution, and updated it with this year’s evidence.

### 2.5.1 Conditioning after state transformation

After all these examples we develop some general results. We start with a fundamental result about conditioning of a transformed state. It can be reformulated as a combination of forward and backward reasoning. This has many consequences.

**Theorem 2.5.7.** Let $c: X \rightarrow Y$ be a channel with a state $\omega \in D(X)$ on its domain and a factor $q \in \text{Fact}(Y)$ on its codomain. Then:

$$(c \gg \omega)|_q = c|_q \gg (\omega|_c \ll q). \quad (2.7)$$
2.5. Reasoning along channels

Proof. For each $y \in Y$,

$$(c \gg \omega)_\eta(y) = \frac{(c \gg \omega)(y) \cdot q(y)}{c \gg \omega \models q} = \sum_x \frac{\omega(x) \cdot c(x)(y) \cdot q(y)}{\omega \models c \ll q} = \sum_x \frac{\omega(x) \cdot (c \ll q)(x) \cdot c(x)(y) \cdot q(y)}{\omega \models c \ll q} = \sum_x \frac{\omega|_{c \ll q}(x) \cdot c(x)(y) \cdot q(y)}{\omega(x) \models q} = \sum_x \frac{\omega|_{c \ll q}(x) \cdot c(x)|_\eta(y)}{c(x) \models q} = \sum_x \omega|_{c \ll q}(x) \cdot c|_\eta(x)(y) = (c|_\eta \gg (\omega|_{c \ll q}))(y).$$

This result has a number of useful consequences.

Corollary 2.5.8. For appropriately typed channels, states, and factors:

1. $(d \circ c)|_\eta = d|_\eta \circ c|_{d \ll q}$;
2. $((c, d) \gg \omega)|_{p \otimes q} = (c|_p, d|_q) \gg \omega|_{c \ll p \otimes q \ll d \ll q}$;
3. $((e \otimes f) \gg \omega)|_{p \otimes q} = (e|_p \otimes f|_q) \gg \omega|_{e \ll p \otimes f \ll q}$. 
Chapter 2. Predicates and Observables

Proof. 1. \((d \circ c)|_q(x) = (d \circ c)(x)|_q\)
   \[d|_q \gg (c(x)|_q) = (d|_q \circ c|_q)(x).\]

2. \((c \circ d) \gg \omega)|_{p \otimes q}\)
   \[= (c|_{p \otimes q} \gg (d|_{p \otimes q} \circ \omega))\]
   by Exercises 2.3.9 and 2.4.6

3. Using the same exercises we also get:
   \[((e \otimes f) \gg \omega)|_{p \otimes q}\]
   \[= ((e|_{p}) \otimes (f|_{q}) \gg ((d|_{q} \otimes \omega) \circ (e|_{p} \otimes f|_{q}))\]
   \[= (e|_{p} \otimes f|_{q}) \gg ((d|_{q} \otimes \omega) \circ (e|_{p} \otimes f|_{q})).\]

Earlier we have seen ‘crossover update’ for a joint state, whereby one updates in one component, and then marginalises in the other, see for instance Example 2.3.2. We shall do this below for joint states \(gr(\sigma, c) = \langle id, c \rangle \gg \sigma\) that arise as graph, see Definition 1.8.10 via a channel. The result below appeared in [13], see also [53, 54]. It says that crossover updates on joint graph states can also be done via forward and backward inferences. This is also a consequence of Theorem 2.5.7.

Corollary 2.5.9. Let \(c: X \Rightarrow Y\) be a channel with a state \(\sigma \in D(X)\) on its domain. Then:

1. for a factor \(p\) on \(X\),
   \[((id, c) \gg \sigma)|_{p \otimes q}[0, 1] = c \gg (\sigma|_{q}).\]

2. for a factor \(q\) on \(Y\),
   \[((id, c) \gg \sigma)|_{p \otimes q}[1, 0] = \sigma|_{c \ll q}.\]

Proof. Since:

1. \((id, c) \gg \sigma)|_{p \otimes q}[0, 1]\)
   \[= \pi_2 \gg (id, c)|_{p \otimes q} \gg (\sigma|_{id \ll c \gg c \ll 1})\]
   \[= (\pi_2 \gg (id, c)) \gg (\sigma|_{id \ll c \gg c \ll 1})\]
   \[= c \gg (\sigma|_{p})\]

2. \((id, c) \gg \sigma)|_{p \otimes q}[1, 0]\)
   \[= \pi_1 \gg (id, c)|_{p \otimes q} \gg (\sigma|_{id \ll c \gg c \ll 1})\]
   \[= (\pi_1 \gg (id, c)) \gg (\sigma|_{id \ll c \gg c \ll 1})\]
   \[= \sigma|_{c \ll q}.\]

This result will tell us how to do inference in a Bayesian network, see Section 3.10.
2.5. Reasoning along channels

Exercises

2.5.1 We consider some disease with an \textit{a priori} probability of 1%. There is a test for the disease with the following ‘sensitivity’. If someone has the disease, then the test is 90% positive; but if someone does not have the disease, there is still a 5% chance that the test is positive.

1. Take as disease space \( D = \{d, d'\} \); describe the prior as a distribution on \( D \);
2. Take as test space \( T = \{p, n\} \) and describe the sensitivity as a channel \( s: D \rightarrow T \);
3. Show that the predicted positive test probability is almost 6%.
4. Assume that a test comes out positive. Use backward reasoning to prove that the probability of having the disease (the posterior) is then a bit more than 15% (to be precise: \( \frac{18}{107} \)). Explain why it is so low — remembering Example 2.5.3.

2.5.2 Give a channel-based analysis and answer to the following question from [87, Chap. I, Exc. 39].
Stores \( A, B, \) and \( C \) have 50, 75, and 100 employees, and respectively, 50, 60, and 70 percent of these are women. Resignations are equally likely among all employees, regardless of sex. One employee resigns and this is a woman. What is the probability that she works in store \( C \)?

2.5.3 Check that the ‘piranha’ predicate \( P: M(\{(p, g)\}) \rightarrow [0, 1] \) in Example 2.5.5 can also be described via the draw function \( D \) from Definition 1.6.3 as:

\[
P(\sigma) = D(\sigma) \models 1_p \otimes 1.
\]

2.5.4 The following situation about the relationship between eating hamburgers and having Kreuzfeld-Jacob disease is inspired by [5, §1.2]. We have two sets: \( E = \{H, H'\} \) about eating Hamburgers (or not), and \( D = \{K, K'\} \) about having Kreuzfeld-Jacob disease (or not). The following distributions on these sets are given: half of the people eat hamburgers, and only one in hundred thousand have Kreuzfeld-Jacob disease, which we write as:

\[
\omega = \frac{1}{2}|H\rangle + \frac{1}{2}|H'\rangle \quad \text{and} \quad \sigma = \frac{1}{100,000}|K\rangle + \frac{99,999}{100,000}|K'\rangle.
\]

1. Suppose that we know that 90% of the people who have Kreuzfeld-Jacob disease eat Hamburgers. Use this additional information to define a channel \( c: D \rightarrow E \) with \( c \gg \sigma = \omega \).
2. Compute the probability of getting Kreuzfeld-Jacob for someone eating hamburgers (via backward inference).
2.5.5 Let \( c : X \to Y \) be a channel with state \( \omega \in \mathcal{D}(X) \) and factors \( p \in \text{Fact}(X) \), \( q \in \text{Fact}(Y) \). Check that:

\[
cl_q \gg (\omega |_{p \land (c \ll q)}) = (c \gg (\omega |_p))|_q.
\]

Describe what this means for \( p = 1 \).

2.5.6 Let \( p \in \text{Fact}(X) \) and \( q \in \text{Fact}(Y) \) be two factors on spaces \( X, Y \).

1 Show that for two channels \( c : Z \to X \) and \( d : Z \to Y \) with a joint state on their (common) domain \( \sigma \in \mathcal{D}(Z) \) one has:

\[
((c, d) \gg \sigma)|_{p \otimes q}[1, 0] = (c \gg (\sigma |_{d \ll q}))[p]
\]

\[
((c, d) \gg \sigma)|_{p \otimes q}[0, 1] = (d \gg (\sigma |_{c \ll p}))[q].
\]

2 For channels \( e : U \to X \) and \( d : V \to Y \) with a joint state \( \omega \in \mathcal{D}(U \times V) \) one has:

\[
((e \otimes f) \gg \omega)|_{p \otimes q}[1, 0] = (e \gg (\omega |_{1 \otimes f \ll q}))[1, 0])|_p
\]

\[
((e \otimes f) \gg \omega)|_{p \otimes q}[0, 1] = (f \gg (\omega |_{e \ll p \otimes q})[0, 1])|_q.
\]

2.6 Discretisation, and coin bias learning

So far we have been using discrete probability distributions, with a finite domain. We shall be looking at continuous distributions later on, in Chapter ??.

In the meantime we can approximate continuous distributions by discretisation, namely by chopping them up into finitely many parts, like in Riemann integration. This section first introduces this discretisation and then uses it in an extensive example on learning the bias of a coin via successive backward inference.

We start with discretisation.

**Definition 2.6.1.** Let \( a, b \in \mathbb{R} \) with \( a < b \) and \( N \in \mathbb{N} \) with \( N > 0 \) be given.

1 We write \([a, b]_N \subseteq [a, b] \subseteq \mathbb{R}\) for the interval \([a, b]\) reduced to \( N \) elements:

\[
[a, b]_N := \{a + \frac{i}{s}, a + \frac{i+1}{s}, \ldots, a + \frac{N-1}{s}\} \quad \text{where} \quad s := \frac{b-a}{N} = \{a + (i + \frac{1}{s})s \mid i \in \mathbb{N}\}.
\]

2 Let \( f : S \to \mathbb{R}_{\geq 0} \) be a function, defined on a finite subset \( S \subseteq \mathbb{R} \). We write \( \text{Disc}(f, S) \in \mathcal{D}(S) \) for the discrete distribution defined as:

\[
\text{Disc}(f, S) := \sum_{x \in S} \frac{f(x)}{t} | \cdot x \rangle \quad \text{where} \quad t := \sum_{x \in S} f(x).
\]
Often we combine the notations from these two points and use discretised states of the form $\text{Disc}(f, [a, b]_N)$.

To see an example of point (1), consider the interval $[1, 2]$ with $N = 3$. The step size $s$ is then $s = \frac{2-1}{3} = \frac{1}{3}$, so that:

$$[1, 2]_3 = \{ 1 + \frac{1}{3}, 1 + \frac{2}{3}, 1 + \frac{4}{3} \} = \{ 1 + \frac{1}{6}, 1 + \frac{1}{2}, 1 + \frac{5}{6} \}$$

We choose to use internal points only and exclude the end-points in this finite subset since the end-points sometimes give rise to boundary problems, with functions being undefined. When $N$ goes to infinity, the smallest and largest elements in $[a, b]_N$ will approximate the end-points $a$ and $b$ — from above and from below, respectively.

The ‘total’ number $t$ in point (2) normalises the formal sum and ensures that the multiplicities add up to one. In this way we can define a uniform distribution on $[a, b]_N$ as $\nu_{[a, b]_N}$, like before, or alternatively as $\text{Disc}(1, [a, b]_N)$, where $1$ is the constant-one function.

**Example 2.6.2.** We look at the following classical question: suppose we are given a coin with an unknown bias, and we observe the following list of heads (1) and tails (0):

$$[0, 1, 1, 0, 0, 1, 1].$$

What can we then say about the bias of the coin?

The frequentist approach that we have seen in Section 1.6 would turn the above list into a multiset and then into a distribution, by frequentist learning, see Diagram (1.10). This would give:

$$[0, 1, 1, 0, 0, 1, 1] \mapsto 5|1\rangle + 3|0\rangle \mapsto \frac{5}{8}|1\rangle + \frac{3}{8}|0\rangle.$$

Here we do not use this frequentist approach to learning the bias parameter, but take a Bayesian route. We assume that the bias parameter itself is given by a distribution, describing the likelihoods of various bias values. We assume no prior knowledge and therefore start from the uniform distribution. It will be updated based on successive observations, using the technique of backward inference, see Definition 2.5.1 (2).

The bias $b$ of a coin is a number in the unit interval $[0, 1]$, giving rise to a coin distribution $\text{flip}(b) = b|1\rangle + (1 - b)|0\rangle$. Thus we can see $\text{flip}$ as a channel $\text{flip}: [0, 1] \rightarrow 2 = \{0, 1\}$. At this stage we avoid continuous distributions and discretise the unit interval. We choose $N = 100$ in the chop up, giving as underlying space $[0, 1]_N$ with $N$ points, on which we take the uniform distribution as prior:

$$\omega := \text{Disc}(1, [0, 1]_N) = \sum_{x \in [0, 1]_N} \frac{1}{N}|x\rangle.$$
We use the \emph{flip} operation as channel, restricted to the discretised space:

\[
[0, 1]_N \xrightarrow{\text{flip}} 2 \quad \text{given by} \quad \text{flip}(b) = b|1\rangle + (1 - b)|0\rangle.
\]

We write \text{yes, no}: 2 \to [0, 1] for the (sharp) predicates \text{yes} = 1_I = \text{id} and \text{no} = \text{yes}^\perp = 1_0. Given the above sequence of head/tail observations [0, 1, 1, 1, 0, 0, 1, 1], we perform successive backward inferences, mapping a 1 in the sequence to an update with predicate \text{flip} \ll \text{yes}, and a 0 to update with \text{flip} \ll \text{no}, as in:

\[
\omega|\text{flip} \ll \text{no} = \omega|\text{flip} \ll \text{no}|\text{yes} = \omega|\text{flip} \ll \text{no}|\text{yes}|\text{yes} = \omega|\text{flip} \ll \text{no}|\text{yes}|\text{yes}|\text{yes} = \omega|\text{flip} \ll \text{no}|\text{yes}|\text{yes}|\text{yes}|\text{yes} = \cdots
\]

An overview of the resulting distributions is given in Figure 2.6. These distributions approximate (continuous) beta distributions, technically because “beta is conjugate prior to flip”, see Section ?? or e.g. [47]. These beta functions form a smoothed out version of the bar charts in Figure 2.6. We shall look more systematically into ‘learning along a channel’ in Section 4.8, where the current coin-bias computation re-appears in Example 4.8.4.

In the end, after these eight updates, let’s write \(\rho = \omega|\text{flip} \ll \text{no}|\text{yes}|\text{yes}|\text{yes}|\text{yes} \cdots \text{for the resulting distribution. We now ask two questions.}

1 What is the predicted coin distribution? The outcome, with truncated multiplicities, is:

\[
\text{flip} \gg \rho = 0.6|1\rangle + 0.4|0\rangle.
\]

2 What is the expected value of \(\rho\). Here we get:

\[
\text{mean}(\rho) = 0.6.
\]

For mathematical reasons\footnote{The distribution \(\rho\) is an approximation of the probability density function \(\beta(6, 4)\), which has mean \(\frac{6}{6+4} = 0.6\).} the exact outcome is 0.6. However, we have used approximation via discretisation. The value computed with this discretisation is 0.59999985316273. We can conclude that chopping the unit interval up with \(N = 100\) already gives a fairly good approximation.

We conclude with a number of observations. The distribution that is obtained by successive backward inferences is determined by the number of 0’s and 1’s in the list of observations, and not by their order. This is because the order of conditioning does not matter. Hence it makes more sense to use multisets
2.6. Discretisation, and coin bias learning

Figure 2.2 Coin bias distributions arising from the prior uniform distribution by successive updates after coin observations [0, 1, 1, 1, 0, 1, 1].

\( n_1 \mid 1 \rangle + n_0 \mid 0 \rangle \in M(2) \) as observations (data), instead of lists, where \( n_1 \in \mathbb{N} \) is the observed number of head’s and \( n_0 \) the number of tails. In the above example we have \( n_1 = 5 \) and \( n_0 = 3 \). Such abstractions from data that keep only the information that is relevant for updating are called sufficient statistics.

**Exercises**

2.6.1 Recall the the \( N \)-element set \([a, b]_N\) from Definition 2.6.1.

1. Show that its largest element is \( b - \frac{1}{2} s \), where \( s = \frac{b - a}{N} \). \( \sum_{i \in \mathbb{N}} i = \frac{N(N-1)}{2} \).

2. Prove that \( \sum_{x \in [a, b]_N} x = \frac{N(a+b)}{2} \).

2.6.2 Consider coin parameter learning in Example 2.6.2.

1. Show that the prediction in the prior (uniform) state \( \omega \) on \([0, 1]_N\) gives a fair coin, i.e.

\[
\text{flip} \gg \omega = \frac{1}{2} \mid 1 \rangle + \frac{1}{2} \mid 0 \rangle.
\]

This equality is independent of \( N \).

2. Prove that, also independently of \( N \),

\[
\omega \mid \text{flip} \ll \text{yes} = \frac{1}{2}.
\]

3. Show next that:

\[
\omega_{\text{flip-yes}} = \sum_{x \in [0,1]_N} \frac{2^x}{N} \mid x \rangle.
\]
4 Use the ‘square pyramidal’ formula \( \sum_{i \in \mathbb{N}} i^2 = \frac{N(N+1)(2N+1)}{6} \) to prove that:

\[
(flip \gg (\omega|_{flip=\text{yes}}))(1) = \frac{2}{3} - \frac{1}{6N^2}.
\]

Conclude that \( flip \gg (\omega|_{flip=\text{yes}}) \) approaches \( \frac{2}{3} |1\rangle + \frac{1}{3} |0\rangle \) as \( N \) goes to infinity.

2.6.3 Prove that the probabilities \((flip \gg \rho)(1)\) and \(\text{mean}(\rho)\) are the same, in the two points at the end of Example 2.6.2.

2.7 Inference in Bayesian networks

In previous sections we have seen several examples of channel-based inference, in forward and backward form. This section shows how to apply these inference methods to Bayesian networks, via an example that is often used in the literature: the ‘Asia’ Bayesian network, originally from [64]. It captures the situation of patients with a certain probability of smoking and of an earlier visit to Asia; this influences certain lung diseases and the outcome of an x-ray test. Later on, in Section 3.10, we shall look more systematically at inference in Bayesian networks.

The Bayesian network example considered here is described in several steps: Figure 2.3 gives the underlying graph structure, in the style of Figure 1.4, introduced at the end of Section 1.9, with diagonals and types of wires written explicitly. Figure 2.4 gives the conditional probability tables associated with the nodes of this network. The way in which these tables are written is different from Section 1.9: we now only write the probabilities \( r \in [0, 1] \) and omit the values \( 1 - r \). Next, these conditional probability tables are described in Figure 2.5 as states smoke \( \in \mathcal{D}(S) \) and asia \( \in \mathcal{D}(A) \) and as channels lung: \( S \rightarrow L \), tub: \( A \rightarrow T \), bronc: \( S \rightarrow B \), xray: \( E \rightarrow X \), dysp: \( B \times E \rightarrow D \), either: \( L \times T \rightarrow E \).

Our aim in this section is illustrate channel-based inference in this Bayesian network. It is not so much the actual outcomes that we are interested in, but more the systematic methodology that is used to obtain these outcomes.

Probability of lung cancer, given no bronchitis

Let’s start with the question: what is the probability that someone has lung cancer, given that this person does not have bronchitis. The latter information is the evidence. It takes the form of a singleton predicate \( 1_{b\perp} = (1_b)^\perp : B \rightarrow [0, 1] \) on the set \( B = \{b, b\perp\} \) used for presence and absence of bronchitis.

In order to obtain this updated probability of lung cancer we ‘follow the
2.7. Inference in Bayesian networks

Figure 2.3 The graph of the Asia Bayesian network, with node abbreviations: bronc = bronchitis, dysp = dyspnea, lung = lung cancer, tub = tuberculosis. The wires all have 2-element sets of the form $A = \{a, a^\perp\}$ as types.

In Figure 2.3 we see that we can transform (pull back) the evidence along the bronchitis channel $\text{bronc} : A \rightarrow B$, and obtain a predicate $\text{bronc} \ll 1_b \perp$ on $A$. The latter can be used to update the asia distribution on $A$. Subsequently, we can push the updated distribution forward along the lung channel $\text{lung} : A \rightarrow L$ via state transformation. Thus we follow the ‘V’ shape in the relevant part of the graph.

Combining this down-update-up steps gives the required outcome:

$$\text{lung} \gg (\text{asia} |_{\text{bronc} \ll 1_b \perp}) = 0.0427(\ell) + 0.9573(\ell^\perp).$$

We see that this calculation combines forward and backward inference, see Definition 2.5.1.

Probability of smoking, given a positive xray

In Figures 2.4 and 2.5 we see a prior smoking probability of 50%. We like to know what this probability becomes if we have evidence of a positive xray. The latter is given by the point predicate $1_x \in \text{Pred}(X)$ for $X = \{x, x^\perp\}$.

Now there is a long path from xray to smoking, see Figure 2.3 that we need to use for (backward) predicate transformation. Along the way there is a complication, namely that the node ‘either’ has two parent nodes, so that pulling back along the either channel yields a predicate on the product set $L \times T$. 

109
Chapter 2. Predicates and Observables

The only sensible thing to do is continue predicate transformation down downwards, but now with the parallel product channel $\text{lung} \otimes \text{tub} : S \times A \rightarrow L \times T$. The resulting predicate on $S \times A$ can be used to update the product state $\text{smoke} \otimes \text{asia}$. Then we can take the first marginal to obtain the desired outcome. Thus we compute:

$$\left( (\text{smoke} \otimes \text{asia}) \big|_{\text{lung} \otimes \text{tub} = \text{either} \otimes \text{xray} = \text{1}_L} \right)[1, 0] = \left( (\text{smoke} \otimes \text{asia}) \big|_{\text{xray} = \text{either} \otimes \text{lung} \otimes \text{tub} = \text{1}_L} \right)[1, 0] = 0.6878 | s \rangle + 0.3122 | s^\perp \rangle. $$

Thus, a positive xray makes it more likely — w.r.t. the uniform prior — that the patient smokes — as is to be expected. This is obtained by backward inference.

**Probability of lung cancer, given both dyspnoea and tuberculosis**

Our next inference challenge involves two evidence predicates, namely $\text{1}_D$ on $D$ for dyspnoea and $\text{1}_T$ on $T$ for tuberculosis. We would like to know the update lung cancer probability.

The situation looks complicated, because of the ‘closed loop’ in Figure 2.3. But we can proceed in a straightforward manner and combine evidence via conjunction & at a suitable meeting point. We now more clearly separate the forward and backward stages of the inference process. We first move the prior

\begin{align*}
\begin{array}{c|c|c}
\text{bronc} & \text{either} & \text{P(dysp)} \\
\text{b} & e & 0.9 \\
\text{b} & e^\perp & 0.7 \\
\text{b^\perp} & e & 0.8 \\
\text{b^\perp} & e^\perp & 0.1 \\
\end{array}
& \quad \\
\begin{array}{c|c|c}
\text{lung} & \text{tub} & \text{P(either)} \\
\ell & t & 1 \\
\ell & t^\perp & 1 \\
\ell^\perp & t & 1 \\
\ell^\perp & t^\perp & 0 \\
\end{array}
\end{align*}

Figure 2.4 The conditional probability tables of the Asia Bayesian network
The result that we are after is now obtained via updating and marginalisation: \[ \sigma \] channels can be ‘shifted’. In particular, in the definition of the above state on the set \( B \times S \times S \). Inference in Bayesian networks

Recall that we write \( \Delta \) for the copy channel, in this expression of type \( S \rightarrow S \times S \).

Going in the backward direction we can form a predicate, called \( p \) below, on the set \( B \times L \times T \), by predicate transformation and conjunction:

\[ p := (1 \otimes 1 \otimes 1) \& ((\text{id} \otimes \text{either}) \ll (\text{dys} \ll 1_d)). \]

The result that we are after is now obtained via updating and marginalisation:

\[ \sigma|_p[0, 0, 1] = 0.6878|\ell) + 0.3122|\ell^c). \quad (2.8) \]

There is an alternative way to describe the same outcome, using that certain channels can be ‘shifted’. In particular, in the definition of the above state \( \sigma \), the channel bronc is used for state transformation. It can also be used in a different role, namely for predicate transformation. We then use a slightly different state, now on \( S \times L \times T \).

\[ \tau := (\text{id} \otimes \text{lung} \otimes \text{id}) \gg ((\Delta \otimes \text{tub}) \gg (\text{smoke} \otimes \text{asia})). \]

The bronc channel is now used for predicate transformation in the predicate:

\[ q := (1 \otimes 1 \otimes 1) \& ((\text{bronc} \otimes \text{either}) \ll (\text{dys} \ll 1_d)). \]

The same updated lung cancer distribution is now obtained as:

\[ \tau|_q[0, 0, 1] = 0.0558|\ell) + 0.9442|\ell^c). \quad (2.9) \]

| smoke | 0.5|s) + 0.5|x^+ | asia | 0.01|a) + 0.99|a^+ |
|--------|------------------|--------|------------------|
| lung(s) | 0.1|\ell) + 0.5|\ell^c | tub(a) | 0.05|t) + 0.95|t^c |
| lung(s^+) | 0.3|\ell) + 0.7|\ell^c | tub(a^+) | 0.01|t) + 0.99|t^c |
| bronc(s) | 0.6|b) + 0.4|b^+ | xray(e) | 0.98|x) + 0.02|x^+ |
| bronc(s^+) | 0.3|b) + 0.7|b^+ | xray(e^+) | 0.05|x) + 0.95|x^+ |
| dysp(b, e) | 0.9|d) + 0.1|d^+ | either(\ell, t) | 1|e |
| dysp(b, e^+) | 0.7|d) + 0.3|d^+ | either(\ell^c, t) | 1|e |
| dysp(b^+, e) | 0.8|d) + 0.2|d^+ | either(\ell, t^c) | 1|e |
| dysp(b^+, e^+) | 0.1|d) + 0.9|d^+ | either(\ell^c, t^c) | 1|e^+ |

Figure 2.5 The conditional probability tables from Figure 2.4 described as states and channels.
The reason why the outcomes (2.8) and (2.9) are the same is the topic of Exercise 2.7.2.

We conclude that inference in Bayesian networks can be done compositionally via a combination of forward and backward inference, basically by following the network structure.

Exercises

2.7.1 Consider the wetness Bayesian network from Section 1.9. Write down the channel-based inference formulas for the following inference questions and check the outcomes that are given below.

1. The updated sprinkler distribution, given evidence of a slippery road, is \( \frac{63}{260} b + \frac{197}{260} \).
2. The updated wet grass distribution, given evidence of a slippery road, is \( \frac{334}{5200} d + \frac{851}{5200} d' \).

(Answers will appear later, in Example 3.3.3, in two different ways.)

2.7.2 Check that the equality of the outcomes in (2.8) and in (2.9) can be explained via Exercises 2.3.5 (2) and 2.4.7.

2.8 Validity-based distances

There are standard notions of distance between states, and also between predicates. It turns out these distances can both be formulated in terms of validity \( \models \), in a dual form. For two states \( \omega_1, \omega_2 \in \mathcal{D}(X) \), we take the join of the distance in \([0,1]\) of validities:

\[
d(\omega_1, \omega_2) := \bigvee_{p \in \text{Pred}(X)} | \omega_1 \models p \models \omega_2 |. \tag{2.10}
\]

Similarly, for two predicates \( p_1, p_2 \in \text{Pred}(X) \),

\[
d(p_1, p_2) := \bigvee_{\omega \in \mathcal{D}(X)} | \omega \models p_1 \models \omega \models p_2 |. \tag{2.11}
\]

This section collects some standard results about these distances (for discrete probability), in order to have a useful overview. We note that the above formulations involve predicates only, not observables in general.
2.8. Validity-based distances

2.8.1 Distance between states

The distance defined in (2.10) is commonly called the total variation distance, which is a special case of the Kantorovich distance, see e.g. [31] [10] [71] [67]. Its characterisations below are standard. We refer to [52] for more information about the validity-based approach.

Proposition 2.8.1. Let \( X \) be an arbitrary set, with states \( \omega_1, \omega_2 \in D(X) \). Then:

\[
d(\omega_1, \omega_2) = \max_{U \subseteq X} \omega_1 \models 1_U - \omega_2 \models 1_U = \frac{1}{2} \sum_{x \in X} |\omega_1(x) - \omega_2(x)|.
\]

We write maximum ‘\( \max \)’ instead of join \( \vee \) to express that the supremum is actually reached by a subset (sharp predicate).

Proof. Let \( \omega_1, \omega_2 \in D(X) \) be two discrete probability distributions on the same set \( X \). We will prove the two inequalities labeled \((a)\) and \((b)\) in:

\[
\frac{1}{2} \sum_{x \in X} |\omega_1(x) - \omega_2(x)| \leq \max_{U \subseteq X} \omega_1 \models 1_U - \omega_2 \models 1_U \leq \bigvee_{p \in \text{Pred}(X)} |\omega_1 \models p - \omega_2 \models p| \leq \frac{1}{2} \sum_{x \in X} |\omega_1(x) - \omega_2(x)|.
\]

This proves Proposition 2.8.1 since the inequality in the middle is trivial.

We start with some preparatory definitions. Let \( U \subseteq X \) be an arbitrary subset. We shall write \( \omega_1(U) = \sum_{x \in U} \omega_1(x) = (\omega \models 1_U) \). We partition \( U \) in three disjoint parts, and take the relevant sums:

\[
\begin{aligned}
U_\geq &= \{x \in U \mid \omega_1(x) > \omega_2(x)\} \\
U_= &= \{x \in U \mid \omega_1(x) = \omega_2(x)\} \\
U_\leq &= \{x \in U \mid \omega_1(x) < \omega_2(x)\}
\end{aligned}
\]

\[
\begin{aligned}
U^+ &= \omega_1(U_\geq) - \omega_2(U_\geq) \geq 0 \\
U^- &= \omega_1(U_\leq) - \omega_2(U_\leq) \geq 0.
\end{aligned}
\]

We use this notation in particular for \( U = X \). In that case we can use:

\[
1 = \omega_1(X) = \omega_1(X_\geq) + \omega_1(X_=) + \omega_1(X_\leq) \\
1 = \omega_2(X) = \omega_2(X_\geq) + \omega_2(X_=) + \omega_2(X_\leq)
\]

Hence by subtraction we obtain, since \( \omega_1(X_\geq) = \omega_2(X_\geq) \),

\[
0 = (\omega_1(X_\geq) - \omega_2(X_\geq)) + (\omega_1(X_\leq) - \omega_2(X_\leq))
\]

That is,

\[
X^+ = \omega_1(X_\geq) - \omega_2(X_\geq) = \omega_2(X_\geq) - \omega_1(X_\geq) = X^-.
\]
Chapter 2. Predicates and Observables

As a result:
\[ \frac{1}{2} \sum_{x \in \mathcal{X}} |\omega_1(x) - \omega_2(x)| \]
\[ = \frac{1}{2} \left( \sum_{x \in \mathcal{X}_+, (\omega_1(x) - \omega_2(x)) + \sum_{x \in \mathcal{X}_- (\omega_2(x) - \omega_1(x))} \right) \]
\[ = \frac{1}{2} \left( (\omega_1(X_+) - \omega_2(X_+)) + (\omega_2(X_-) - \omega_1(X_-)) \right) \tag{2.12} \]
\[ = \frac{1}{2} (X_\uparrow + X_\downarrow) \]
\[ = X_\uparrow \]

We have prepared the ground for proving the above inequalities (a) and (b).

(a) We will see that the above maximum is actually reached for the subset \( U = X_\uparrow \), first of all because:
\[ \frac{1}{2} \sum_{x \in \mathcal{X}} |\omega_1(x) - \omega_2(x)| \]
\[ = \omega_1 \uparrow X_\uparrow = \omega_1(X_+) - \omega_2(X_+) \]
\[ = \omega_1 \uparrow 1 - \omega_2 \uparrow 1 \]
\[ \leq \max_{U \subseteq \mathcal{X}} \omega_1 \uparrow 1 - \omega_2 \uparrow 1. \]

(b) Let \( p \in \text{Pred}(X) \) be an arbitrary predicate. We have: \( (1_U \land p)(x) = 1_U(x) \cdot p(x) \), which is \( p(x) \) if \( x \in U \) and 0 otherwise. Then:
\[ \omega_1 \uparrow p - \omega_2 \uparrow p \]
\[ = \left( \omega_1 \uparrow 1_x \land p + \omega_1 \uparrow 1_x \land p \right) \]
\[ - (\omega_2 \uparrow 1_x \land p + \omega_2 \uparrow 1_x \land p) \]
\[ = \left( \omega_1 \uparrow 1_x \land p - \omega_2 \uparrow 1_x \land p \right) \]
\[ - (\omega_2 \uparrow 1_x \land p - \omega_1 \uparrow 1_x \land p) \]
\[ \leq \begin{cases} \omega_1 \uparrow 1_x \land p - \omega_2 \uparrow 1_x \land p & \text{if } (\ast) \\ \omega_2 \uparrow 1_x \land p - \omega_1 \uparrow 1_x \land p & \text{otherwise} \end{cases} \]
\[ = \begin{cases} \sum_{x \in \mathcal{X}_+ (\omega_1(x) - \omega_2(x)) \cdot p(x) & \text{if } (\ast) \\ \sum_{x \in \mathcal{X}_- (\omega_2(x) - \omega_1(x)) \cdot p(x) & \text{otherwise} \end{cases} \]
\[ \leq \begin{cases} \sum_{x \in \mathcal{X}_+ (\omega_1(x) - \omega_2(x)) \cdot p(x) & \text{if } (\ast) \\ \sum_{x \in \mathcal{X}_- (\omega_2(x) - \omega_1(x)) & \text{otherwise} \end{cases} \]
\[ = X_\uparrow \uparrow \text{ if } (\ast) \]
\[ = X_\downarrow = X_\uparrow \text{ otherwise.} \]
2.8. Validity-based distances

This completes the proof.

The sum-formulation in Proposition 2.8.1 is useful in many situations, for instance in order to prove that the above distance function \( d \) between states is a metric.

Lemma 2.8.2. The distance \( d(\omega_1, \omega_2) \) between states \( \omega_1, \omega_2 \in \mathcal{D}(X) \) in \( (2.10) \) turns the set of distributions \( \mathcal{D}(X) \) into a metric space, with \([0, 1]\)-valued metric.

Proof. If \( d(\omega_1, \omega_2) = \frac{1}{2} \sum_{x \in X} |\omega_1(x) - \omega_2(x)| = 0 \), then \( |\omega_1(x) - \omega_2(x)| = 0 \) for each \( x \in X \), so that \( \omega_1(x) = \omega_2(x) \), and thus \( \omega_1 = \omega_2 \). Obviously, \( d(\omega_1, \omega_2) = d(\omega_2, \omega_1) \). The triangle inequality holds for \( d \) since it holds for the standard distance on \([0, 1]\).

\[
\begin{align*}
d(\omega_1, \omega_2) &= \frac{1}{2} \sum_{x \in X} |\omega_1(x) - \omega_2(x)| \\
&\leq \frac{1}{2} \sum_{x \in X} |\omega_1(x) - \omega_2(x)| + |\omega_2(x) - \omega_3(x)| \\
&= \frac{1}{2} \sum_{x \in X} |\omega_1(x) - \omega_2(x)| + \frac{1}{2} \sum_{x \in X} |\omega_2(x) - \omega_3(x)| \\
&= d(\omega_1, \omega_2) + d(\omega_2, \omega_3). \\
\end{align*}
\]

We use this same sum-formulation for the following result.

Lemma 2.8.3. State transformation is non-expansive: for a channel \( c : X \rightarrow Y \) one has:

\[
d(c \gg \omega_1, c \gg \omega_2) \leq d(\omega_1, \omega_2).
\]

Proof. Since:

\[
\begin{align*}
d(c \gg \omega_1, c \gg \omega_2) &= \frac{1}{2} \sum_y |(c \gg \omega_1)(y) - (c \gg \omega_2)(y)| \\
&= \frac{1}{2} \sum_x \left| \sum_y c(x)(y) \cdot \omega_1(x) - \sum_y c(x)(y) \cdot \omega_2(x) \right| \\
&= \frac{1}{2} \sum_y \left| \sum_x c(x)(y) \cdot (\omega_1(x) - \omega_2(x)) \right| \\
&\leq \frac{1}{2} \sum_y \left| \sum_x c(x)(y) \right| \cdot |\omega_1(x) - \omega_2(x)| \\
&= \frac{1}{2} \sum_x \left( \sum_y c(x)(y) \right) \cdot |\omega_1(x) - \omega_2(x)| \\
&= d(\omega_1, \omega_2). \\
\end{align*}
\]

For two states \( \omega_1, \omega_2 \in \mathcal{D}(X) \) a coupling is a joint state \( \sigma \in \mathcal{D}(X \times X) \) that marginalises to \( \omega_1 \) and \( \omega_2 \), i.e. \( \sigma[1, 0] = \omega_1 \) and \( \sigma[0, 1] = \omega_2 \). Such couplings give an alternative formulation of the distance between states, which is often called the Wasserstein distance, see \([100]\). These couplings will also be used in Section 3.8 as probabilistic relations. The proof of the next result is standard and is included in order to be complete.
Proposition 2.8.4. For states \( \omega_1, \omega_2 \in D(X) \),

\[
d(\omega_1, \omega_2) = \bigwedge \{ \sigma \models \text{Eq}^+ \mid \sigma \text{ is a coupling between } \omega_1, \omega_2 \},
\]

where \( \text{Eq} : X \times X \to [0, 1] \) is the equality predicate from Definition 2.7.1 [4].

**Proof.** We use the notation and results from the proof of Proposition 2.8.1. We first prove the inequality \( \leq \). Let \( X_\ast = \{ x \in X \mid \omega_1(x) > \omega_2(x) \} \). Then:

\[
\omega_1(x) = \sum_x \sigma(x, y) = \sigma(x, x) + \sum_{y \neq x} \sigma(x, y)
\]

\[
\leq \omega_2(x) + (\sigma \models (1 \otimes 1) & \text{Eq}^+).
\]

This means that \( \omega_1(x) - \omega_2(x) \leq \sigma \models (1 \otimes 1) & \text{Eq}^+ \) for \( x \in X_\ast \). We similarly have:

\[
\omega_2(x) = \sum_x \sigma(y, x) = \sigma(x, x) + \sum_{y \neq x} \sigma(y, x)
\]

\[
\leq \omega_1(x) + (\sigma \models (1 \otimes 1) & \text{Eq}^+).
\]

Hence \( \omega_2(x) - \omega_1(x) \) for \( x \notin X_\ast \). Putting this together gives:

\[
d(\omega_1, \omega_2) = \frac{1}{2} \sum_{x \in X_\ast} |\omega_1(x) - \omega_2(x)|
\]

\[
= \frac{1}{2} \sum_{x \in X_\ast} \omega_1(x) - \omega_2(x) + \frac{1}{2} \sum_{x \notin X_\ast} \omega_2(x) - \omega_1(x)
\]

\[
\leq \frac{1}{2} \sum_{x \in X_\ast} \sigma \models (1 \otimes 1) & \text{Eq}^+ + \frac{1}{2} \sum_{x \notin X_\ast} \sigma \models (1 \otimes 1) & \text{Eq}^+
\]

\[
= \frac{1}{2} \sigma \models \text{Eq}^+ + \frac{1}{2} \sigma \models \text{Eq}^+
\]

\[
= \sigma \models \text{Eq}^+.
\]

For the inequality \( \geq \) one uses what is called an optimal coupling \( \rho \in D(X \times X) \) of \( \omega_1, \omega_2 \). It can be defined as:

\[
\rho(x, y) := \begin{cases} 
\min \{ \omega_1(x), \omega_2(x) \} & \text{if } x = y \\
\max \{ \omega_1(x) - \omega_2(x), 0 \} \max \{ \omega_2(y) - \omega_1(y), 0 \} / d(\omega_1, \omega_2) & \text{otherwise}.
\end{cases}
\]

We first check that this \( \rho \) is a coupling. Let \( x \in X_\ast \) so that \( \omega_1(x) > \omega_2(x) \); then:

\[
\sum_y \rho(x, y)
\]

\[
= \omega_2(x) + (\omega_1(x) - \omega_2(x)) \cdot \sum_{y \neq x} \frac{\max \{ \omega_2(y) - \omega_1(y), 0 \}}{d(\omega_1, \omega_2)}
\]

\[
= \omega_2(x) + (\omega_1(x) - \omega_2(x)) \cdot \frac{\sum_{x \in X_\ast} \omega_2(y) - \omega_1(y)}{d(\omega_1, \omega_2)}
\]

\[
= \omega_2(x) + (\omega_1(x) - \omega_2(x)) \cdot \frac{X_\ast}{d(\omega_1, \omega_2)}
\]

\[
= \omega_2(x) + (\omega_1(x) - \omega_2(x)) \cdot 1 \text{ see the proof of Proposition 2.8.1}
\]

\[
= \omega_1(x).
\]
2.8. Validity-based distances

If \( x \not\in X \), so that \( \omega_1(x) \leq \omega_2(x) \), then it is obvious that \( \sum_y \rho(x, y) = \omega_1(x) + 0 = \omega_1(x) \). This shows \( \sigma[1, 0] = \omega_1 \). In a similar way one obtains \( \sigma[0, 1] = \omega_2 \).

Finally,

\[
\rho \models Eq = \sum_x \rho(x, x) = \sum_x \min (\omega_1(x), \omega_2(x)) = \sum_{x \in X} \omega_2(x) + \sum_{x \in X} \omega_1(x) = \omega_2(X) + 1 - \omega_1(X).
\]

This shows \( \sigma[1, 0] = \omega_1 \). In a similar way one obtains \( \sigma[0, 1] = \omega_2 \).

Finally,

\[
\rho \models Eq = \sum_x \rho(x, x) = \sum_x \min (\omega_1(x), \omega_2(x)) = \sum_{x \in X} \omega_2(x) + 1 - \omega_1(X).
\]

Hence \( d(\omega_1, \omega_2) = 1 - (\rho \models Eq) = \rho \models Eq^c \).

2.8.2 Distance between predicates

What we have to say about the validity-based distance (2.11) between predicates is rather brief. First, there is also a point-wise formulation.

**Lemma 2.8.5.** For two predicates \( p_1, p_2 \in \text{Pred}(X) \),

\[
d(p_1, p_2) = \bigvee_{x \in X} |p_1(x) - p_2(x)|.
\]

This distance function \( d \) makes the set \( \text{Pred}(X) \) into a metric space.

**Proof.** First, we have for each \( x \in X \),

\[
d(p_1, p_2) \overset{\text{TT}}{=} \bigvee_{\omega \in \Omega(X)} |\omega \models p_1 - \omega \models p_2| \geq |\text{unit}(x) \models p_1 - \text{unit}(x) \models p_2| = |p_1(x) - p_2(x)|.
\]

Hence \( d(p_1, p_2) \geq \bigvee_x |p_1(x) - p_2(x)| \).

The other direction follows from:

\[
|\omega \models p_1 - \omega \models p_2| = |\sum_z \omega(z) \cdot p_1(z) - \sum_z \omega(z) \cdot p_2(z)| \leq \sum_z \omega(z) \cdot |p_1(z) - p_2(z)| \leq \sum_z \omega(z) \cdot \bigvee_{x \in X} |p_1(x) - p_2(x)|.
\]

\[
= (\sum_z \omega(z)) \cdot \bigvee_{x \in X} |p_1(x) - p_2(x)|.
\]

The fact that we get a metric space is now straightforward.

We have the analogue of Lemma 2.8.3.
Lemma 2.8.6. Predicate transformation is non-expansive: for a channel \( c: X \rightarrow Y \) one has, for predicates \( p_1, p_2 \in \text{Pred}(Y) \),
\[
d(c \ll p_1, c \ll p_2) \leq d(p_1, p_2).
\]

Proof. Via the formulation of Lemma 2.8.5 we get:
\[
d(c \ll p_1, c \ll p_2) = \bigvee_x |(c \ll p_1)(x) - (c \ll p_2)(x)|
\]
\[
= \bigvee_x |\sum_y c(x)(y) \cdot p_1(y) - \sum_y c(x)(y) \cdot p_2(y)|
\]
\[
\leq \bigvee_x |\sum_y (c(x)(y)) \cdot |p_1(y) - p_2(y)||
\]
\[
\leq \bigvee_x |\sum_y (c(x)(y)) \cdot d(p_1, p_2)|
\]
\[
= d(p_1, p_2).
\]

Exercises

2.8.1 In \([54]\) the influence of a predicate \( p \) on a state \( \omega \) is measured via the distance \( d(\omega, \omega \gg p) \). This influence can be zero, for the truth predicate \( p = 1 \).

Consider the set \( \{H, T\} \) with state \( \sigma(\varepsilon) = \varepsilon|H\rangle + (1-\varepsilon)|T\rangle \), and with predicate \( p = 1_H \). Prove that \( d(\sigma(\varepsilon), \sigma(\varepsilon)\gg p) \to 1 \) as \( \varepsilon \to 0 \).

2.8.2 This exercise uses the distance between a joint state and the product of its marginals as measure of entwinedness, like in \([44]\).

1 Take \( \sigma_2 := \frac{1}{2}|00\rangle + \frac{1}{2}|11\rangle \in \mathcal{D}(2 \times 2) \), for \( 2 = \{0, 1\} \). Show that:

\[
d(\sigma_2, \sigma_2[1,0] \otimes \sigma_2[0,1]) = \frac{1}{2}.
\]

2 Take \( \sigma_3 := \frac{1}{2}|000\rangle + \frac{1}{2}|111\rangle \in \mathcal{D}(2 \times 2 \times 2) \). Show that:

\[
d(\sigma_3, \sigma_3[1,0,0] \otimes \sigma_3[0,1,0] \otimes \sigma_3[0,0,1]) = \frac{3}{4}.
\]

3 Now define \( \sigma_n \in \mathcal{D}(2^n) \) for \( n \geq 2 \) as:

\[
\sigma_n := \frac{1}{2^n} \left( \begin{array}{c}
0 \cdots 0 \\
\vdots \\
1 \cdots 1
\end{array} \right).
\]

Show that:

- each marginal \( \pi_i \gg \sigma_n \) equals \( \frac{1}{2^n} |0\rangle + \frac{1}{2^n} |1\rangle \);
- the product \( \otimes_i (\pi_i \gg \sigma_n) \) of the marginals is the uniform state on \( 2^n \);
- \( d(\sigma_n, \otimes_i (\pi_i \gg \sigma_n)) = \frac{2^{n-1} - 1}{2^{n-1}} \).
Note that the latter distance goes to 1 as \( n \) goes to infinity.

2.8.3 Show that \( D(X) \), with the total variation distance (2.8), is a complete metric space when the set \( X \) is finite.

### 2.9 Variance, covariance and correlation

This section describes the standard notions of variance, covariance and correlation within the setting of this book. It uses the validity relation \( \models \) and the operations on observables from Section 2.2.2. Recall that we understand a random variable here as a pair \((\omega, p)\) consisting of a state \(\omega \in D(X)\) and an observable \(p: X \to \mathbb{R}\). It turns out that it is highly relevant which combination of state and observable is used: the notions of covariance and correlation are defined for two random variables. The question then comes up: do these two random variables involve the same state, like in \((\omega, p_1)\) and \((\omega, p_2)\), or do we have two random variables with different states, like in \((\omega_1, p_1)\) and \((\omega_2, p_2)\), or is there an underlying joint state? These differences are significant, but are usually not dealt with explicitly when the state associated with a random variable is left implicit.

Let \((\omega, p)\) be a random variable on a set \(X\). The validity \(\omega \models p\) is a real number, and can thus be used as a scalar, in the sense of Section 2.2.2. The truth predicate forms an observable \(1 \in \text{Obs}(X)\); scalar multiplication yields a new observable \((\omega \models p) \cdot 1 \in \text{Obs}(X)\). It can be subtracted\(^3\) from \(p\), and then squared, giving an observable:

\[
(p - (\omega \models p) \cdot 1)^2 = (p - (\omega \models p) \cdot 1) \& (p - (\omega \models p) \cdot 1) \in \text{Obs}(X).
\]

This observable denotes the function that sends \(x \in X\) to \((p(x) - (\omega \models p))^2 \in \mathbb{R}\). Its validity in the original state \(\omega\) is called variance. It captures how far the values of \(p\) differ from their expected value.

**Definition 2.9.1.** For a random variable \((\omega, p)\), the variance \(\text{Var}(\omega, p)\) is the non-negative number defined by:

\[
\text{Var}(\omega, p) := \omega \models (p - (\omega \models p) \cdot 1)^2.
\]

This number is non-negative since it is the validity of a factor: the right-hand-side of \(\models\) is a square and is thus \(\mathbb{R}_{\geq 0}\)-valued.

\(^3\) Subtraction expressions like these occur more frequently in mathematics. For instance, an eigenvalue \(\lambda\) of a matrix \(M\) may be defined as the scalar that forms a solution to the equation \(M - \lambda \cdot 1 = 0\), where \(1\) is the identity matrix. A similar expression is used to define the elements in the spectrum of a \(C^*\)-algebra. See also Exercise 2.2.8.
When the underlying sample space $X$ is a subset of $\mathbb{R}$, say via incl: $X \hookrightarrow \mathbb{R}$, then one simply writes $\text{Var}(\omega)$ for $\text{Var}(\omega, \text{incl})$.

The name standard deviation is used for the square root of the variance; thus:

$$\text{StDev}(\omega, p) := \sqrt{\text{Var}(\omega, p)}.$$  

**Example 2.9.2.** 1 We recall Example 2.1.4 (4), with distribution $\text{flip}(0.3) = 0.3(1) + 0.7(0)$ and observable $v(0) = -50$ and $v(1) = 100$. We had $\omega \models v = -5$, and so we get:

$$\text{Var}(\text{flip}(0.3), v) = \sum_x \text{flip}(0.3)(x) \cdot (v(x) + 5)^2$$

$$= 0.3 \cdot (100 + 5)^2 + 0.7 \cdot (-50 + 5)^2 = 4725.$$  

The standard deviation is around 68.7.

2 For a (fair) dice we have $\text{pips} = \{1, 2, 3, 4, 5, 6\} \hookrightarrow \mathbb{R}$ and $\text{mean}(\text{dice}) = \frac{7}{2}$ so that:

$$\text{Var}(\text{dice}) = \sum_x \text{dice}(x) \cdot (x - \frac{7}{2})^2$$

$$= \frac{1}{6} \cdot \left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = \frac{70}{21}.$$  

The following result is sometimes useful in calculations, see the subsequent binomial example.

**Lemma 2.9.3.** Variance satisfies:

$$\text{Var}(\omega, p) = (\omega \models p^2) - (\omega \models p)^2.$$  

**Proof.** We have:

$$\text{Var}(\omega, p)$$

$$= (p - (\omega \models p) \cdot 1)^2$$

$$= \sum_x \omega(x)(p(x) - (\omega \models p))^2$$

$$= \sum_x \omega(x)\left(p(x)^2 - 2(\omega \models p)p(x) + (\omega \models p)^2\right)$$

$$= \left(\sum_x \omega(x)p(x)^2\right) - 2(\omega \models p)\left(\sum_x \omega(x)p(x)\right) + \left(\sum_x \omega(x)(\omega \models p)^2\right)$$

$$= (\omega \models p^2) - 2(\omega \models p)(\omega \models p) + (\omega \models p)^2$$

$$= (\omega \models p^2) - (\omega \models p)^2. \quad \square$$  

We use this result to obtain the variance of a binomial distribution.

**Example 2.9.4.** We fix $K \in \mathbb{N}$ and $r \in [0, 1]$ and use incl: $\{0, 1, \ldots, K\} \hookrightarrow \mathbb{R}$ as observable. We recall from Example 2.1.4 (4) that:

$$\text{mean}(\text{binom}[K](r)) = \text{binom}[K](r) \models \text{incl} = K \cdot r.$$  

120
2.9. Variance, covariance and correlation

By re-using the same trick as for the mean calculation, we first compute:

\[
\text{binom}(K)(r) \equiv \text{incl} \& (\text{incl} - 1)
\]

\[
= \sum_{0 \leq k \leq K} \text{binom}(K)(r) \cdot k \cdot (k - 1)
\]

\[
= \sum_{0 \leq k \leq K} \frac{k \cdot (k - 1)}{k! \cdot (K - k)!} \cdot r^k \cdot (1 - r)^{K - k}
\]

\[
= K \cdot (K - 1) \cdot r^2 \cdot \frac{(K - 2)!}{2! \cdot (K - 2)!} \cdot r^2 \cdot (1 - r)^{(K - 2) - (K - 2)}
\]

\[
= K \cdot (K - 1) \cdot r^2 \cdot \sum_{0 \leq j \leq K - 2} \frac{(K - 2)!}{j! \cdot ((K - 2) - j)!} \cdot r^j \cdot (1 - r)^{(K - 2) - j}
\]

\[
= K \cdot (K - 1) \cdot r^2.
\]

Then, using the reformulation of Lemma 2.9.3,

\[
\text{Var(\text{binom}(K)(r))}
\]

\[
= (\text{binom}(K)(r) \equiv \text{incl} \& \text{incl}) - (\text{binom}(K)(r) \equiv \text{incl})^2
\]

\[
= (\text{binom}(K)(r) \equiv \text{incl} \& (\text{incl} - 1)) + (\text{binom}(K)(r) \equiv \text{incl}) - (K \cdot r)^2
\]

\[
= K \cdot (K - 1) \cdot r^2 + K \cdot r - (K \cdot r)^2
\]

\[
= K \cdot r \cdot (1 - r).
\]

We then notice that variance of the binomial is ‘symmetric’ in the probability, in the sense that \(\text{Var(binom}(K)(r)) = \text{Var(binom}(K)(1 - r))\).

We continue with covariance and correlation, which involve two random variables, instead one, as for variance. We distinguish two forms of these, namely:

- The two random variables are of the form \((\omega, p_1)\) and \((\omega, p_2)\), where they share their state \(\omega\).
- There is a joint state \(\tau \in \mathcal{D}(X_1 \times X_2)\) together with two observables \(q_1 : X_1 \to \mathbb{R}\) and \(q_2 : X_2 \to \mathbb{R}\) on the two components \(X_1, X_2\). This situation can be seen as a special case of the previous point by first weakening the two observables to the product space, via: \(\pi_1 \ll q_1 = q_1 \otimes 1\) and \(\pi_2 \ll q_2 = 1 \otimes q_2\). Thus we obtain two random variable with a shared state:

\[
(\tau, \pi_1 \ll q_1) \quad \text{and} \quad (\tau, \pi_2 \ll q_2).
\]

These observable transformations \(\pi_i \ll q_i\) along a deterministic channel \(\pi_i : X_1 \times X_2 \to X_i\) can also be described simply as function composition \(q_i \circ \pi_i : X_1 \times X_2 \to \mathbb{R}\), see Lemma 2.4.2.[7]

We start with the situation in the first bullet above, and deal with the second bullet in Definition 2.9.3.
Chapter 2. Predicates and Observables

Definition 2.9.5. Let \((\omega, p_1)\) and \((\omega, p_2)\) be two random variable with a shared state \(\omega \in \mathcal{D}(X)\).

1. The **covariance** of these random variables is defined as the validity:

\[
\text{Cov}(\omega, p_1, p_2) := \omega \models (p_1 - (\omega \models p_1) \cdot 1) \& (\omega \models p_2 - (\omega \models p_2) \cdot 1).
\]

2. The **correlation** between \((\omega, p_1)\) and \((\omega, p_2)\) is the covariance divided by their standard deviations:

\[
\text{Cor}(\omega, p_1, q_2) := \frac{\text{Cov}(\omega, p_1, p_2)}{\text{StDev}(\omega, p_1) \cdot \text{StDev}(\omega, p_2)}.
\]

Notice that variance is a special case of covariance, namely where \(p = p_1 = p_2\). Hence if there is an inclusion \(\text{incl}: X \hookrightarrow \mathbb{R}\) and we would use this inclusion twice to compute covariance, we are in fact computing variance.

Before we go on, we state the following analogue of Lemma 2.9.3, leaving the proof to the reader.

Lemma 2.9.6. Covariance can be reformulated as:

\[
\text{Cov}(\omega, p_1, p_2) = (\omega \models p_1 \& p_2) - (\omega \models p_1) \cdot (\omega \models p_2).
\]

Example 2.9.7. 1 We have seen in Definition 2.1.2 (2) how the average of an observable can be computed as its validity in a uniform state. The same approach is used to compute the covariance (and correlation) in a uniform joint state. Consider the following to lists \(a\) and \(b\) of numerical data, of the same length.

\[a = [5, 10, 15, 20, 25] \quad b = [10, 8, 10, 15, 12]\]

We will identify \(a\) and \(b\) with random variables, namely with \(a, b: 5 \rightarrow \mathbb{R}\), where \(5 = \{0, 1, 2, 3, 4\}\). Hence there are obvious definitions: \(a(0) = 5, a(1) = 10, a(2) = 15, a(3) = 20, a(4) = 25\), and similarly for \(b\). Then we can compute their averages as validities in the uniform state \(\nu_5\) on the set \(5\):

\[
\text{avg}(a) = \nu_5 \models a = 15 \quad \text{avg}(a) = \nu_5 \models a = 11.
\]

We will calculate the covariance between \(a\) and \(b\) wrt. the uniform state \(\nu_5\), as:

\[
\text{Cov}(\nu_5, a, b) = \nu_5 \models (a - (\nu_5 \models a) \cdot 1) \& (b - (\nu_5 \models b) \cdot 1)
\]

\[
= \sum_i \left( \frac{1}{5} \cdot (a(i) - 15) \cdot (b(i) - 11) \right)
\]

\[
= 11.
\]
2 In order to obtain the correlation between $a, b$, we first need to compute their variances:

\[
\begin{align*}
\text{Var}(v_5, a) &= \sum_i \frac{1}{2} (a(i) - 15)^2 = 50 \\
\text{Var}(v_5, b) &= \sum_i \frac{1}{2} (b(i) - 11)^2 = 5.6
\end{align*}
\]

Then:

\[
\text{Cor}(v_5, a, b) = \frac{\text{Cov}(v_5, a, b)}{\sqrt{\text{Var}(v_5, a)} \cdot \sqrt{\text{Var}(v_5, b)}} = \frac{11}{\sqrt{50} \cdot \sqrt{5.6}} \approx 0.66.
\]

We will next define covariance and correlation for a joint state with observables on its component spaces, as a special case of what we have seen so far. We shall refer to this instance as joint covariance and joint correlation, with notation $\text{JCov}$ and $\text{JCor}$.

**Definition 2.9.8.** Let $\tau \in D(X_1 \times X_2)$ be a joint state on sets $X_1, X_2$ and let $q_1 \in \text{Obs}(X_1)$ and $q_2 \in \text{Obs}(X_2)$ be two observables on these two sets separately. In this situation the joint covariance is defined as:

\[
\text{JCov}(\tau, q_1, q_2) := \text{Cov}(\tau, \pi_1 \ll q_1, \pi_2 \ll q_2).
\]

Similarly, the joint correlation is:

\[
\text{JCor}(\tau, q_1, q_2) := \text{Cor}(\tau, \pi_1 \ll q_1, \pi_2 \ll q_2).
\]

In both these cases, if there are inclusions $X_1 \hookrightarrow \mathbb{R}$ and $X_2 \hookrightarrow \mathbb{R}$, then one can use these inclusions as random variables and write just $\text{JCov}(\tau)$ and $\text{JCor}(\tau)$.

Joint covariance can be reformulated in different ways, including in the style that we have seen before, in Lemma 2.9.3 and 2.9.6.

**Lemma 2.9.9.** Joint covariance can be reformulated as:

\[
\text{JCov}(\tau, q_1, q_2) = \tau \models (q_1 - (\tau[1, 0] \models q_1) \cdot 1) \otimes (q_2 - (\tau[0, 1] \models q_2) \cdot 1).
\]

**Proof.** The first equation follows from:

\[
\begin{align*}
\text{JCov}(\tau, q_1, q_2) &= \text{Cov}(\tau, \pi_1 \ll q_1, \pi_2 \ll q_2).
\end{align*}
\]

\[
= \tau \models ((\pi_1 \ll q_1) - (\tau \models \pi_1 \ll q_1) \cdot 1)
\]

\[
&\quad \& (\pi_2 \ll q_2) - (\tau \models \pi_2 \ll q_2) \cdot 1)
\]

\[
= \tau \models ((\pi_1 \ll q_1) - (\tau[1, 0] \models q_1) \cdot (\pi_1 \ll 1))
\]

\[
&\quad \& ((\pi_2 \ll q_2) - (\tau[0, 1] \models q_2) \cdot (\pi_2 \ll 1))
\]

\[
= \tau \models (\pi_1 \ll (q_1 - (\tau[1, 0] \models q_1) \cdot 1))
\]

\[
&\quad \& (\pi_2 \ll (q_2 - (\tau[0, 1] \models q_2) \cdot 1))
\]

\[
= \tau \models (q_1 - (\tau[1, 0] \models q_1) \cdot 1) \otimes (q_2 - (\tau[0, 1] \models q_2) \cdot 1).
\]
In order to calculate the (joint) correlation of $\tau_1$ consider sets $\text{Example 2.9.10.}$ In the joint case it makes sense to leave these inclusions implicit, as will be joint form, especially if the underlying sets are subsets of the real numbers.

\[ JCov(\tau, q_1, q_2) = \tau \models (q_1 - (\tau [1, 0] \models q_1) \cdot 1) \otimes (q_2 - (\tau [0, 1] \models q_2) \cdot 1) \]
\[ = \sum_{x,y} \tau (x,y) \cdot (q_1 - (\tau [1, 0] \models q_1) \cdot 1)(x) \cdot (q_2 - (\tau [0, 1] \models q_2) \cdot 1)(y) \]
\[ = \sum_{x,y} \tau (x,y) \cdot (q_1(x) - (\tau [1, 0] \models q_1)) \cdot (q_2(y) - (\tau [0, 1] \models q_2)) \]
\[ = (\sum_{x,y} \tau (x,y) \cdot q_1(x) \cdot q_2(y)) - (\sum_{x,y} \tau (x,y) \cdot q_1(x) \cdot (\tau [0, 1] \models q_2)) \]
\[ - (\sum_{x,y} \tau (x,y) \cdot (\tau [1, 0] \models q_1) \cdot q_2(y)) + (\tau [1, 0] \models q_1)(\tau [0, 1] \models q_2) \]
\[ = (\tau \models q_1 \otimes q_2) - (\tau [1, 0] \models q_1)(\tau [0, 1] \models q_2). \]

We turn to some illustrations. Many examples of covariance are actually of joint form, especially if the underlying sets are subsets of the real numbers. In the joint case it makes sense to leave these inclusions implicit, as will be illustrated below.

**Example 2.9.10.** 1 Consider sets $X = \{1, 2\}$ and $Y = \{1, 2, 3\}$ as subsets of $\mathbb{R}$, together with a joint distribution $\tau \in \mathcal{D}(X \times Y)$ given by:

\[ \tau = \frac{1}{2}[1, 1] + \frac{1}{2}[1, 2] + \frac{1}{4}[2, 2] + \frac{1}{4}[2, 3]. \]

Its two marginals are:

\[ \tau [1, 0] = \frac{1}{2}[1] + \frac{1}{2}[2] \quad \tau [0, 1] = \frac{1}{4}[1] + \frac{1}{4}[2] + \frac{1}{4}[3]. \]

Since both $X \hookrightarrow \mathbb{R}$ and $Y \hookrightarrow \mathbb{R}$ we get means as validities of the inclusions:

\[ \text{mean}(\tau [1, 0]) = \frac{3}{4} \quad \text{mean}(\tau [1, 0]) = 2. \]

Now we can compute the joint covariance in the joint state $\tau$ as:

\[ JCov(\tau) = \tau \models (\text{incl} - \text{mean}(\tau [1, 0]) \cdot 1) \otimes (\text{incl} - \text{mean}(\tau [1, 0]) \cdot 1) \]
\[ = \sum_{x,y} \tau (x,y) \cdot (x - \frac{3}{2}) \cdot (y - 2) \]
\[ = \frac{1}{4}(-\frac{3}{2} \cdot -1 + \frac{1}{2} \cdot 1) \]
\[ = \frac{3}{4}. \]

2 In order to calculate the (joint) correlation of $\tau$ we first need the variances of its marginals:

\[ \text{Var}(\tau [1, 0]) = \sum_x \tau [1, 0](x) \cdot (x - \frac{3}{2})^2 = \frac{1}{4} \]
\[ \text{Var}(\tau [0, 1]) = \sum_y \tau [0, 1](y) \cdot (y - 2)^2 = \frac{1}{2}. \]

Then:

\[ JCor(\tau) = \frac{JCov(\tau)}{\sqrt{\text{Var}(\tau [1, 0]) \cdot \sqrt{\text{Var}(\tau [0, 1])}}} = \frac{\frac{3}{4}}{\frac{1}{2} \cdot \frac{1}{\sqrt{2}}} = \frac{1}{2} \sqrt{2}. \]
2.9. Variance, covariance and correlation

We have defined joint covariance as a special case of ordinary covariance. We conclude this section by showing that ordinary covariance can also be seen as joint covariance, namely for a copied state.

**Proposition 2.9.11.** Ordinary covariance can be expressed as joint covariance:

\[
\text{Cov}(\omega, p_1, p_2) = \text{JCov}(\Delta \gg \omega, p_1, p_2).
\]

**Proof.** We use that the marginals of the copied state \( \Delta \gg \omega \) are \( \omega \) itself, in:

\[
\text{JCov}(\Delta \gg \omega, p_1, p_2)
= \Delta \gg \omega \models (p_1 - ((\Delta \gg \omega)[1, 0] \models p_1) \cdot 1)
\otimes (p_2 - ((\Delta \gg \omega)[0, 1] \models p_2) \cdot 1)
\text{ by Lemma 2.9.9}
= \omega \models (p_1 - (\omega \models p_1) \cdot 1) \otimes (p_2 - (\omega \models p_2) \cdot 1)
\text{ by Exercise 2.4.5 (2)}
= \omega \models (p_1 - (p_1 - (\omega \models p_1) \cdot 1) \cdot 1) \otimes (p_2 - (p_2 - (\omega \models p_2) \cdot 1) \cdot 1)
= \omega \models \text{Cov}(\omega, p_1, p_2).
\]

We started with ‘ordinary’ covariance in Definition 2.9.5 for two random variables with a shared state. It was subsequently used to define the ‘joint’ version in Definition 2.9.8. The above result shows that we could have done this the other way around: obtain the ordinary formulation in terms of the joint version. In the light of this result we shall also refer to the ordinary version of covariance as the ‘copied state’ formulation — because of the role of copying in \( \Delta \gg \omega \). The same terminology appears in Section 4.7 in the context of learning from data.

**Exercises**

2.9.1 Let \( \tau \in \mathcal{D}(X \times Y) \) be a joint state with an observable \( p : X \to \mathbb{R} \). We can turn them into a random variable in two ways, by marginalisation and weakening:

\[
(\tau[1, 0], p) \quad \text{and} \quad (\tau, \pi_1 \ll p).
\]

where \( \pi_1 \ll p = p \otimes 1 \) is an observable on \( X \times Y \).

These two random variables have the same expected value, by (2.4). Show that they have the same variance too:

\[
\text{Var}(\tau[1, 0], p) = \text{Var}(\tau, \pi_1 \ll p).
\]

As a result, the standard deviations are also the same.
2.9.2 Prove that:

\[ J\text{Cor}(\tau, q_1, q_2) = \frac{J\text{Cov}(\tau, q_1, q_2)}{\text{StDev}(\tau[1, 0], q_1) \cdot \text{StDev}(\tau[0, 1], q_2)}. \]

2.9.3 Prove Lemma 2.9.6 along the lines of the proof of Lemma 2.9.3.

2.9.4 Find examples of covariance and correlation computations in the literature (or online) and determine if they are of copied-state or joint-state form.

2.10 Dependence and covariance

This section explains the notion of dependence between random variables in the current setting. We shall follow the approach of the previous section and introduce two versions of dependence, also called copied-state and joint-state. Independence is related to the property ‘covariance is zero’, but in a subtle way. This will be elaborated below.

Suppose we have two random variable describing the number of icecream sales and the temperature. Intuitively one expects a dependence between the two, and that the two variables are correlated (in an informal sense). The opposite, namely independence is usually formalised as follows. Two random variables \( p_1, p_2 \) are called independent if:

\[ P[p_1 = a, p_2 = b] = P[p_1 = a] \cdot P[p_2 = b]. \] (2.13)

holds for all real numbers \( a, b \). In this formulation a distribution is assumed in the background. We like to use it explicitly. How should the above equation (2.13) then be read?

As we described in Subsection 2.1.1, the expression \( P[p = a] \) can be interpreted as transformation along the observable \( p_1 \), considered as deterministic channel:

\[ P[p = a] = \text{transfo } p \models \omega \models 1_a. \]

where \( \omega \) is the implicit distribution.

We can then translate the above equation (2.13) into:

\[ \langle p_1, p_2 \rangle \models \omega = (p_1 \models \omega) \otimes (p_2 \models \omega). \] (2.14)

But this says that the joint state \( \langle p_1, p_2 \rangle \models \omega \) on \( \mathbb{R} \times \mathbb{R} \), transformed along \( \langle p_1, p_2 \rangle : X \to \mathbb{R} \times \mathbb{R} \), is non-entwined: it is the product of its marginals. Indeed,
2.10. Dependence and covariance

its (first) marginal is:

\[
\langle (p_1, p_2) \rangle \gg \omega[1, 0] = \pi_1 \gg \langle (p_1, p_2) \rangle \gg \omega
\]

\[
= (\pi_1 \otimes \langle p_1, p_2 \rangle) \gg \omega
\]

\[
= p_1 \gg \omega.
\]

This brings us to the following formulation. We formulate it for two random variables, but it can easily be extended to \(n\)-ary form.

**Definition 2.10.1.** Let \((\omega, p_1)\) and \((\omega, p_2)\) be two random variables with a common state \(\omega\). They will be called **independent** if the transformed state

\[
\langle (p_1, p_2) \rangle \gg \omega = (p_1 \otimes p_2) \gg (\Delta \gg \omega)
\]

on \(\mathbb{R} \times \mathbb{R}\) is non-entwined, as in Equation (2.14): it is the product of its marginals.

We sometimes call this the *copied-state* form of independence, in order to distinguish it from a later *joint-state* version.

We give an illustration, of non-independence, that is, of dependence.

**Example 2.10.2.** Consider a fair coin flip \(= \frac{1}{2} |1\rangle + \frac{1}{2} |0\rangle\). We are going to use it twice, first to determine how much we will bet (either €100 or €50), and secondly to determine whether the bet is won or not. Thus we use the distribution \(\omega = \text{flip} \otimes \text{flip}\) with underlying set \(2 \times 2\), where \(2 = \{0, 1\}\). We will define two observables \(p_1, p_2: 2 \times 2 \rightarrow \mathbb{R}\) on this same distribution \(\omega\).

We first define an auxiliary observable \(p: 2 \rightarrow \mathbb{R}\) for the amount of the bet:

\[
p(1) = 100 \quad \quad p(0) = 50.
\]

We then define \(p_1 = p \otimes 1 = p \circ \pi_1: 2 \times 2 \rightarrow \mathbb{R}\), as on the left below. The random variable \(p_2\) is defined on the right.

\[
p_1(x, y) := \begin{cases} 100 & \text{if } x = 1 \\ 50 & \text{if } x = 0 \end{cases} \quad \quad p_2(x, y) := \begin{cases} p(x) & \text{if } y = 1 \\ -p(x) & \text{if } y = 0. \end{cases}
\]

We claim that \((\omega, p_1)\) and \((\omega, p_2)\) are not independent. Intuitively this may be clear, since the observable \(p\) forms a connection between \(p_1\) and \(p_2\). Formally, we can see this by doing the calculations:

\[
\langle (p_1, p_2) \rangle \gg \omega = \mathcal{D}(\langle p_1, p_2 \rangle)(\text{flip} \otimes \text{flip})
\]

\[
= \frac{1}{4} \langle p_1(1, 1), p_2(1, 1) \rangle + \frac{1}{4} \langle p_1(1, 0), p_2(1, 0) \rangle
\]

\[
+ \frac{1}{4} \langle p_1(0, 1), p_2(0, 1) \rangle + \frac{1}{4} \langle p_1(0, 0), p_2(0, 0) \rangle
\]

\[
= \frac{1}{4} 100, 100 \rangle + \frac{1}{4} 100, -100 \rangle + \frac{1}{4} 50, 50 \rangle + \frac{1}{4} 50, -50 \rangle.
\]
The two marginals are:

\[
p_1 \gg \omega = ((p_1, p_2) \gg \omega)[1, 0] = \frac{1}{4}[100] + \frac{1}{4}[50] \\
p_2 \gg \omega = ((p_1, p_2) \gg \omega)[0, 1] = \frac{1}{4}[100] + \frac{1}{4}[100] + \frac{1}{4}[50] + \frac{1}{4}[1] - 50.
\]

It is not hard to see that the parallel product \(\otimes\) of these two marginal distributions differs from \((p_1, p_2) \gg \omega\).

**Proposition 2.10.3.** The copied-state covariance of copied-state independent random variables is zero: if random variables \((\omega, p_1)\) and \((\omega, p_2)\) are independent, then \(\text{Cov}(\omega, p_1, p_2) = 0\).

The converse does not hold.

**Proof.** If \((\omega, p_1)\) and \((\omega, p_2)\) are independent, then, by definition, \((p_1, p_2) \gg \omega = (p_1 \gg \omega) \otimes (p_2 \gg \omega)\). The calculation belows shows that covariance is then zero. It uses multiplication \&: \(\mathbb{R} \times \mathbb{R} \to \mathbb{R}\) as observable. It can also be described as the parallel product \(\text{id} \otimes \text{id}\) of the observable \(\text{id}: \mathbb{R} \to \mathbb{R}\) with itself.

\[
\text{Cov}(\omega, p_1, p_2) \\
= (\omega \models p_1 & \omega \models p_2) - (\omega \models p_1) \cdot (\omega \models p_2) \quad \text{by Lemma 2.9.6} \\
= (\omega \models & \circ \langle p_1, p_2 \rangle) \\
- (p_1 \gg \omega \models \text{id}) \cdot (p_2 \gg \omega \models \text{id}) \quad \text{by Exercise 2.1.3} \\
= ((p_1, p_2) \gg \omega \models \text{id}) \\
- ((p_1 \gg \omega) \otimes (p_2 \gg \omega) \models \text{id} \otimes \text{id}) \quad \text{by Exercises 2.2.5, 2.1.3} \\
= ((p_1, p_2) \gg \omega \models \text{id}) - ((p_1, p_2) \gg \omega \models \text{id}) \quad \text{by assumption} \\
= 0.
\]

The claim that the converse does not hold follows from Example 2.10.4 right below.

**Example 2.10.4.** We continue Example 2.10.2. The set-up used there involves two dependent random variables \((\omega, p_1)\) and \((\omega, p_2)\), with shared state \(\omega = \text{flip} \otimes \text{flip}\). We show here that they (nevertheless) have covariance zero. This proves the second claim of Proposition 2.10.3, namely that zero-covariance need not imply independence, in the copied-state context.

We first compute the validities (expected values):

\[
\omega \models p_1 = \frac{1}{4} \cdot p_2(1, 1) + \frac{1}{4} \cdot p_1(1, 0) + \frac{1}{4} \cdot p_1(0, 1) + \frac{1}{4} \cdot p_1(0, 0) \\
= \frac{1}{4} \cdot 100 + \frac{1}{4} \cdot 100 + \frac{1}{4} \cdot 50 + \frac{1}{4} \cdot 100 = 75 \\
\omega \models p_2 = \frac{1}{4} \cdot 100 + \frac{1}{4} \cdot -100 + \frac{1}{4} \cdot 50 + \frac{1}{4} \cdot -50 = 0.
\]
Then:
\[
\text{Cov}(\omega, p_1, p_2) = \omega \mid \equiv (p_1 \equiv (\omega \equiv p_1) \cdot 1) \& (p_2 \equiv (\omega \equiv p_2) \cdot 1)
\]
\[
= \frac{1}{4}((100-75) \cdot 100 + 100-75) \cdot -100 + (50-75) \cdot 50 + (50-75) \cdot -50
\]
\[
= 0.
\]

We now turn to independence in joint-state form, in analogy with joint-state covariance in Definition 2.9.8.

**Definition 2.10.5.** Let \( \tau \in \mathcal{D}(X_1 \times X_2) \) be a joint state and with two observables \( q_1 \in \text{Obs}(X_1) \) and \( q_2 \in \text{Obs}(X_2) \). We say that there is joint-state independence of \( q_1, q_2 \) if the two random variables \( (\tau, \pi_1 \ll q_1) \) and \( (\tau, \pi_2 \ll q_2) \) are (copied-state) independent, as described in Definition 2.10.1.

Concretely, this means that:
\[
(q_1 \otimes q_2) \gg \tau = (q_1 \gg \tau[1,0]) \otimes (q_2 \gg \tau[0,1]).
\] (2.15)

Equation (2.15) is an instance of the formulation (2.14) used in Definition 2.10.1 since:
\[
\langle q_1 \circ \pi_1, q_2 \circ \pi_2 \rangle \gg \tau = \mathcal{D}((q_1 \circ \pi_1, q_2 \circ \pi_2))(\tau)
\]
\[
= \mathcal{D}(q_1 \times q_2)(\tau)
\]
\[
= (q_1 \otimes q_2) \gg \tau
\]
\[
((q_1 \circ \pi_1) \gg \tau) \otimes ((q_2 \circ \pi_2) \gg \tau) = (q_1 \gg (\pi_1 \gg \tau)) \otimes (q_2 \gg (\pi_2 \gg \tau))
\]
\[
= (q_1 \gg \tau[1,0]) \otimes (q_2 \gg \tau[0,1]).
\]

In the joint-state case — unlike in the copied state situation — there is a tight connection between non-entwinedness, independence and covariance being zero.

**Theorem 2.10.6.** For a joint state \( \tau \) the following three statements are equivalent.

1. \( \tau \) is non-entwined, i.e. is the product of its marginals;
2. all observables \( q_i \in \text{Obs}(X_i) \) are binary independent wrt. \( \tau \);
3. the joint-state covariance \( JCov(\tau, q_1, q_2) \) is zero, for all observables \( q_i \in \text{Obs}(X_i) \) — or equivalently, all correlations \( JCor(\tau, q_1, q_2) \) are zero.

**Proof.** Let joint state \( \tau \in \mathcal{D}(X_1 \times X_2) \) be given. We write \( \tau_i := \pi_i \gg \tau \in \mathcal{D}(X_i) \) for its marginals.

(1) \( \Rightarrow \) (2). If \( \tau \) is non-entwined, then \( \tau = \tau_1 \otimes \tau_2 \). Hence for all observables \( q_1 \in \text{Obs}(X_1) \) and \( q_2 \in \text{Obs}(X_2) \) we have that \( \sigma := (q_1 \otimes q_2) \gg \tau \)
is non-entwined. To see this, first note that \( \pi_i \gg \sigma = q_i \gg \tau_i \). Then, by Lemma 2.10.1:

\[
(\pi_1 \gg \sigma) \otimes (\pi_2 \gg \sigma) = (q_1 \gg \tau_1) \otimes (q_2 \gg \tau_2)
\]

\[
= (q_1 \otimes q_2) \gg (\tau_1 \otimes \tau_2) = (q_1 \otimes q_2) \gg \tau = \sigma.
\]

(2) \( \Rightarrow \) (3). Let \( q_1 \in \text{Obs}(X_1) \) and \( q_2 \in \text{Obs}(X_2) \) be two observables. We may assume that \( q_1, q_2 \) are independent wrt. \( \tau \), that is, \((q_1 \otimes q_2) \gg \tau = (q_1 \gg \tau_1) \otimes (q_2 \gg \tau_2) \) as in (2.15). We must prove \( \text{JCov}(\tau, q_1, q_2) = 0 \). Consider, like in the proof of Proposition 2.10.3, the multiplication map \( \& : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \), given by \( \& (r, s) = r \cdot s \), as an observable on \( \mathbb{R} \times \mathbb{R} \). We consider its validity:

\[
(q_1 \otimes q_2) \gg \tau \models \& = \sum_{r,s} ((q_1 \otimes q_2) \gg \tau)(r,s) \cdot \& (r,s) \\
= \sum_{r,s} D(q_1 \times q_2)(\tau)(r,s) \cdot r \cdot s \\
= \sum_{r,s} (\sum_{x,y} q_1(x)q_2(y)\tau(x,y)) \cdot r \cdot s \\
= \sum_{x,y} \tau(x,y) \cdot (q_1 \otimes q_2)(x,y) \\
= \tau \models q_1 \otimes q_2.
\]

In the same way one proves \((q_1 \gg \tau_1) \otimes (q_2 \gg \tau_2) \models \& = (\tau_1 \models q_1) \cdot (\tau_2 \models q_2) \). But then we are done via the formulation of binary covariance from Lemma 2.9.9

\[
\text{JCov}(\tau, q_1, q_2) = (\tau \models q_1 \otimes q_2) - (\tau_1 \models q_1) \cdot (\tau_2 \models q_2) \\
= ((q_1 \otimes q_2) \gg \tau \models \&) - ((q_1 \gg \tau_1) \otimes (q_2 \gg \tau_2) \models \&) \\
= ((q_1 \otimes q_2) \gg \tau \models \&) - ((q_1 \otimes q_2) \gg \tau \models \&) \\
= 0.
\]

(3) \( \Rightarrow \) (1). Let joint-state covariance \( \text{JCov}(\tau, q_1, q_2) \) be zero for all observables \( q_1, q_2 \). In order to prove that \( \tau \) is non-entwined, we have to show \( \tau(x,y) = \tau_1(x) \cdot \tau_2(y) \) for all \((x,y) \in X_1 \times X_2 \). We choose as random variables the observables \( 1_x \) and \( 1_y \), and use again the formulation of covariance from Lemma 2.9.9

Then, since, by assumption, the binary covariance is zero, so that:

\[
\tau(x,y) = \tau \models 1_x \otimes 1_y = (\tau_1 \models 1_x) \cdot (\tau_2 \models 1_y) = \tau_1(x) \cdot \tau_2(y).
\]

In essence this result says that joint-state independence and joint-state covariance being zero are not properties of observables, but of joint states.

**Exercises**

2.10.1 Prove, in the setting of Definition 2.10.5 that the first marginal \((q_1 \otimes q_2) \gg \tau \) \([1,0]\) of the transformed state along \( q_1 \otimes q_2 \) is equal to the transformed marginal \( q_1 \gg \tau [1,0] \).
In the previous two chapters we have seen several examples of graphs in probabilistic reasoning, see for instance Figures 1.3, 1.4 or 2.3. At this stage a more systematic view will be developed in terms of so-called string diagrams. The latter are directed graphs that are built up inductively from boxes (as nodes) with wires (as edges) between them. These boxes can be interpreted as probabilistic channels. The wires are typed, where a type is a finite set that is associated with a wire. On a theoretical level we make a clear distinction between syntax and semantics, where graphs are introduced as syntactic objects, defined inductively, like terms in some formal language based on a signature or grammar. Graphs can be given a semantics via an interpretation as suitable collections of channels. In practice, the distinction between syntax and semantics is not so sharp: we shall frequently use graphs in interpreted form, essentially by describing a structured collection of channels as interpretation of a graph. This is like what happens in (mathematical) practice with algebraic notions like group or monoid: one can precisely formulate a syntax for monoids with operation symbols that are interpreted as actual functions. But one can also use monoids as sets with operations. The latter form is often most convenient.

String diagrams are similar to the graphs used for Bayesian networks. A notable difference is that they have explicit operations for copying and discarding. This makes them more expressive (and more useful). String diagrams are the language of choice to reason about a large class of mathematical models, called symmetric monoidal categories, see [90], including categories of channels (of powerset, multiset or distribution type). Once we have seen string diagrams, we will describe Bayesian networks only in string diagrammatic form. The most notable difference it that we then write copying as an explicit operation, as we have already started doing in Section 1.9. The language of string diagrams also makes it easier to turn a Bayesian network into a joint distribution.
One of the main themes in this chapter is how to move back and forth between a joint state/distribution, say on $A, B$, and a channel $A \rightarrow B$. The process of extracting a channel form a joint state is called disintegration. It is the process of turning a joint distribution $p(a, b)$ into a conditional distribution $p(b \mid a)$. It gives direction to a probabilistic relationship. It is this disintegration technique that allows us to represent a joint state, on multiple sets, via a graph structure with multiple channels, like in Bayesian networks, by performing disintegration multiple times. More explicitly, it is via disintegration that the conditional probability tables of a Bayesian network — which, recall, correspond to channels — can be obtained from a joint distribution. The latter can be presented as a (possibly large) empirical distribution, analogously to the (small) medicine - blood pressure example in Subsections [1.4.1 and 1.4.5].

Via the correspondence between joint states and channels one can also revert channels. Roughly it works as follows. First one turns a channel $A \rightarrow B$ into a joint state on $A, B$, then into a joint state on $B, A$ by swapping, and then again into a channel $B \rightarrow A$. It amounts to turning a conditional probability $p(b \mid a)$ into $p(a \mid b)$, essentially by Bayes' rule. This channel inversion is a fundamental operation in Bayesian theory, which bears resemblance to inversion (adjoint transpose) for matrices in quantum theory. Since the superscript notation $(\cdot)^\dagger$ is often used in such situations, a ‘dagger’ terminology has become common. It is used for instance in the notion of a ‘dagger’ category, see e.g. [16, 90]. In line with this tradition we often talk about the dagger of a channel, instead of about its Bayesian inversion. This Bayesian inversion is an extremely useful operation, not only in classification, but also in inference. It allows us in Subsection [3.7.1] to establish a tight connection between the direction of a channel and the direction of inference: backward inference along a channel $c$ can be expressed as forward inference along the inverted channel $c^\dagger$, and similarly, forward inference along $c$ can be expressed as backward inference along $c^\dagger$. This fundamental result (Theorem [3.7.7]) gives the channel-based framework internal consistency.

For readers interested in the more categorical aspects of Bayesian inversion, Section [3.8] elaborates how a suitable category of channels has a well-behaved dagger functor. This category of channels is shown to be equivalent to a category of couplings. This mimics the categorical structure underlying the relationship between functions $X \rightarrow \mathcal{P}(Y)$ and relations on $X \times Y$. This categorically-oriented section can be seen as an intermezzo that is not necessary for the remainder; it could be skipped.

Sections [3.9 and 3.10] return to (directed) graphical modelling. Section [3.9]
3.1 String diagrams

uses string diagrams to express intrinsic graphical structure, called shape, in joint probability distributions. This is commonly expressed in terms of independence and d-separation, but we prefer to stick at this stage to a purely string-diagrammatic account. String diagrams are also used in Section 3.10 to give a precise account of Bayesian inference, including a (high-level) channel-based inference algorithm.

The chapter concludes with an alternative form of backward probabilistic reasoning, called Jeffrey’s rule. The approach that we have used up to this point, namely backward inference along a channel, may be called Pearl’s rule, in contrast. It uses a predicate on the codomain of the channel as new information that is used for updating via predicate transformation. Thus, this approach is evidence-based. In contrast, Jeffrey’s rule is state-based: it uses a state on the codomain of the channel and reasons backward via state transformation using the dagger of the channel. These two rules, of Pearl and of Jeffrey, give quite different outcomes in the same situation. It is not well-understood which rule should be used in which situation. Our account focuses on a precise description of these two rules in terms of channels.

3.1 String diagrams

In Chapter 1 we have seen how (probabilistic) channels can be composed sequentially, via \( \circ \cdot \), and in parallel, via \( \otimes \). These operations \( \circ \cdot \) and \( \otimes \) satisfy certain algebraic laws, such as associativity of \( \circ \cdot \). Parallel composition \( \otimes \) is also associative, in a suitable sense, but the equations involved are non-trivial to work with. These equations are formalised via the notion of symmetric monoidal category that axiomatises the combination of sequential and parallel composition.

Instead of working with such somewhat complicated categories we shall work with string diagrams. These diagrams form a graphical language that offers a more convenient and intuitive way of reasoning with sequential and parallel composition. In particular, these diagrams allow obvious forms of topological stretching and shifting that are more complicated, and less intuitive, to capture via equations.

The main reference for string diagrams is [90]. Here we present a more focused account, concentrating on what we need in a probabilistic setting. In this first section we describe string diagrams with directed edges between. In the next section we simply write undirected edges \( | \) instead of directed edges \( \uparrow \), with the convention that information always flows upward. But we have to keep in mind that we do this for convenience only, and that the graphs involved are actually directed.
String diagrams look a bit like Bayesian networks, but differ in subtle ways, esp. wrt. copying, discarding and joint states, see Subsection [3.1.1]. The shift that we are making from Bayesian networks to string diagrams was already prepared in Subsection [1.9.1].

**Definition 3.1.1.** The collection $\text{Type}$ of types is the smallest set with:

1. $1 \in \text{Type}$, where $1$ represents the trivial singleton type;
2. $F \in \text{Type}$ for any finite set $F$;
3. if $A, B \in \text{Type}$, then also $A \times B \in \text{Type}$.

In fact, the first point can easily be seen as a special case of the second one. However, we like to emphasise that there is a distinguished type that we write $1$. More generally, we write, as before, $n = \{0, 1, \ldots, n - 1\}$, so that $n \in \text{Type}$.

The nodes of a string diagram are ‘boxes’ with input and output wires:

![String diagram](image)

Here the $A_1, \ldots, A_n \in \text{Type}$ are the types of the input wires, and $B_1, \ldots, B_m \in \text{Type}$ are the types of the output wires. We allow the cases $n = 0$ (no input wires) and $m = 0$ (no output wires), which are written as:

![String diagram](image) and ![String diagram](image)

A diagram signature, or simply a signature, is a collection of boxes as in (3.1). The interpretation of such a box is a channel $A_1 \times \cdots \times A_n \Rightarrow B_1 \times \cdots \times B_m$.

In particular, the interpretation of a box with no input channels is a distribution/state. We like to make a careful distinction between syntax (boxes in a diagram) and semantics (their interpretation as channels).

Informally, a string diagram is a diagram that is built-up inductively, by connecting wires of the same type between several boxes. These connections may include copying and swapping, see below for details. A string diagram involves the requirement that it contains no cycles: it should be acyclic. We write string diagrams with arrows flow going upwards, suggesting that there is a flow from bottom to top. This direction is purely a convention. Elsewhere in the literature the flow may be from top to bottom, of from left to right.

More formally, given a signature $\Sigma$ of boxes of the form (3.1), the set $\text{SD}(\Sigma)$ of string diagrams over $\Sigma$ is defined as the smallest set satisfying the points below. At the same time we define two ‘domain’ and ‘codomain’ functions
from string diagrams to lists of types, as in:
\[
SD(\Sigma) \xrightarrow{\text{dom}} L(\text{Type}) \xrightarrow{\text{cod}}
\]

Recall that \( L \) is the list functor.

(1) \( \Sigma \subseteq SD(\Sigma) \), that is, every box in \( f \in \Sigma \) in the signature as in (3.1) is a string diagram in itself, with \( \text{dom}(f) = [A_1, \ldots, A_n] \) and \( \text{cod}(f) = [B_1, \ldots, B_m] \).

(2) For all \( A, B \in \text{Type} \), the following diagrams are in \( SD(\Sigma) \).

(a) The identity diagram:
\[
\text{id}_A = A
\]
with \( \begin{cases} \text{dom}(\text{id}_A) = [A] \\ \text{cod}(\text{id}_A) = [A] \end{cases} \)

(b) The swap diagram:
\[
\text{swap}_{A,B} =
\]
with \( \begin{cases} \text{dom}(\text{swap}_{A,B}) = [A, B] \\ \text{cod}(\text{swap}_{A,B}) = [B, A] \end{cases} \)

(c) The copy diagram:
\[
\text{copy}_A =
\]
with \( \begin{cases} \text{dom}(\text{copy}_A) = [A] \\ \text{cod}(\text{copy}_A) = [A, A] \end{cases} \)

(d) The discard diagram:
\[
\text{discard}_A =
\]
with \( \begin{cases} \text{dom}(\text{discard}_A) = [A] \\ \text{cod}(\text{discard}_A) = [] \end{cases} \)

(e) The uniform state diagram:
\[
\text{uniform}_A =
\]
with \( \begin{cases} \text{dom}(\text{uniform}_A) = [] \\ \text{cod}(\text{uniform}_A) = [A] \end{cases} \)

(f) For each type \( A \) and element \( a \in A \) a point state diagram:
\[
\text{point}_{A}(a) =
\]
with \( \begin{cases} \text{dom}(\text{point}_{A}(a)) = [] \\ \text{cod}(\text{point}_{A}(a)) = [A] \end{cases} \)

(g) Boxes for the logical operations conjunction, orthosupplement and scalar multiplication, of the form:
where \( r \in [0, 1] \). These string diagrams have obvious domains and codomains. We recall that predicates on \( A \) can be identified with channels \( A \to 2 \), see Exercise 2.4.8.

(3) If \( S_1, S_2 \in SD(\Sigma) \) are string diagrams, then the \textit{parallel composition} (juxtaposition) \( S_1 S_2 \) is also string diagram, with \( \text{dom}(S_1 S_2) = \text{dom}(S_1) + + \text{dom}(S_2) \) and \( \text{cod}(S_1 S_2) = \text{cod}(S_1) + + \text{cod}(S_2) \).

(4) If \( S_1, S_2 \in SD(\Sigma) \) and \( \text{cod}(S_1) \) ends with \( \vec{C} = [C_1, \ldots, C_k] \) whereas \( \text{dom}(S_2) \) starts with \( \vec{C} \), then there is a \textit{sequential composition} string diagram \( T \) of the form:

\[
T = \begin{array}{c}
\vdots \\
S_2 \\
\vdots \\
\vec{C} \\
\vdots \\
S_1 \\
\vdots 
\end{array}
\quad \text{with} \quad \begin{cases}
\text{dom}(T) = \text{dom}(S_1) + (\text{dom}(S_2) - \vec{C}) \\
\text{cod}(T) = (\text{cod}(S_1) - \vec{C}) + \text{cod}(S_2).
\end{cases}
\]

(3.2)

From this basic form, different forms of composition can be obtained by reordering wires via swapping.

In this definition of string diagrams all wires point upwards. This excludes that string diagrams contain cycles: they are acyclic by construction. One could add more basic constructions to point (2), such as convex combination, see Exercise 3.2.3, but the above constructions suffices for now.

Below we redescribe the ‘wetness’ and ‘Asia’ Bayesian networks, from Fig-
3.1. String diagrams

What are the signatures for these string diagrams?

3.1.1 String diagrams versus Bayesian networks

One can ask: why are these strings diagrams any better than the diagrams that we have seen so far for Bayesian networks? An important point is that string diagrams make it possible to write a joint state, on an $n$-ary product type. For instance, for $n = 2$ we can have a write a joint state as:

$$(3.4)$$

In Bayesian networks joint states cannot be expressed. It is possible to write an initial node with multiple outgoing wires, but as we have seen in Section 1.9, this should be interpreted as copy of a single outgoing wire, coming out of a non-joint state:

$$(3.3)$$

In string diagrams one can additionally express marginals via discarding $\otimes$: if a joint state $\omega$ is the interpretation of the above string diagram (3.4), then the diagram on the left below is interpreted as the marginal $\omega[1,0]$, whereas the
diagram on the right is interpreted as \( \omega[0, 1] \).

We shall use this marginal notation with masks also for boxes in general — and not just for states, without incoming wires. This is in line with Definition 1.8.7 where marginalisation is defined both for states and for channels.

An additional advantage of string diagrams is that they can be used for equational reasoning. This will be explained in the next section. First we give meaning to string diagrams.

### 3.1.2 Interpreting string diagrams as channels

String diagrams will be used as representations of (probabilistic) channels. We thus consider string diagrams as a special, graphical term calculus (syntax). These diagrams are formal constructions, which are given mathematical meaning via their interpretation as channels.

This interpretation of string diagrams can be defined in a compositional way, following the structure of a diagram, using sequential \( \circ \) and parallel \( \otimes \) composition of channels. An interpretation of a string diagram \( S \in SD(\Sigma) \) is a channel that is written as \([ [S] ]\). It forms a channel from the product of the types in the domain \( \text{dom}(S) \) of the string diagram to the product of types in the codomain \( \text{cod}(S) \). The product over the empty list is the singleton element \( 1 = \{0\} \). Such an interpretation \([ [S] ]\) is parametrised by an interpretation of the boxes in the signature \( \Sigma \) as channels.

Given such an interpretation of the boxes in \( \Sigma \), the points below describe this interpretation function \([ [ - ] ]\) inductively, acting on the whole of \( SD(\Sigma) \), by precisely following the above four points in the inductive definition of string diagrams.

- **1.** The interpretation of a box \((3.1)\) in \( \Sigma \) is assumed. It is some channel of type \( A_1 \times \cdots \times A_n \rightarrow B_1 \times \cdots \times B_m \).
- **2.** We use the following interpretation of primitive boxes.
  - (a) The interpretation of the identity wire of type \( A \) is the identity/unit channel \( \text{id} : A \rightarrow A \), given by point states: \( \text{id}(a) = 1|a\rangle \).
  - (b) The swap string diagram is interpreted as the (switched) pair of projections \( \langle \pi_2, \pi_1 \rangle : A \times B \rightarrow B \times A \). It sends \( (a, b) \) to \( 1|b, a\rangle \).
  - (c) The interpretation \([ [\text{copy}_A] ]\) of the copy string diagram is the copy channel \( \Delta : A \rightarrow A \times A \) with \( \Delta(a) = 1|a, a\rangle \).
(d) The discard channel $\triangleright$ of type $A$ is interpreted as the unique channel $!: A \rightarrow 1$. Using that $D(1) \cong 1$, this function sends every element $a \in A$ to the sole element $1|0\rangle$ of $D(1)$ for $1 = |0\rangle$.

(e) The uniform string diagram $\triangleright$ of type $A$ is interpreted as the uniform distribution/state $\nu_A \in D(A)$, considered as channel $1 \rightarrow A$. Recall, if $A$ has $n$-elements, then $\nu_A = \sum_{a \in A} \frac{1}{n}|a\rangle$.

(f) The point string diagram point$_A(a)$, for $a \in A$ is interpreted as the point state $1|a\rangle \in D(A)$.

(g) The ‘logical’ channels for conjunction, orthosupplement, and scalar multiplication are interpreted by the corresponding channels $\text{conj}$, $\text{orth}$ and $\text{scal}(r)$ from Exercise [2.4.8].

(3) The tensor product $\otimes$ of channels is used to interpret juxtaposition of string diagrams $[[S_1 S_2]] = [[S_1]] \otimes [[S_2]]$.

(4) If we have string diagrams $S_1, S_2$ in a composition $T$ as in (3.2), then:

$$[[T]] = (\text{id} \otimes [[S_2]]) \circ ([[[S_1]] \otimes \text{id}),$$

where the identity channels have the appropriate product types.

At this stage we can say more precisely what a Bayesian network is, within the setting of string diagrams.

**Definition 3.1.2.** A Bayesian network is given by:

1. a string diagram $G \in SD(\Sigma)$ over a finite signature $\Sigma$, with $\text{dom}(G) = []$;
2. an interpretation of each box in $\Sigma$ as a channel.

The string diagram $G$ in the first point is commonly referred to as the (underlying) graph of the Bayesian network. The channels that are used as interpretations are the conditional probability tables of the network.

Given the definition of the boxes in (the signature of) a Bayesian network, the whole network can be interpreted as a state/distribution $[[G]]$. This is in general not the same thing as the joint distribution associated with the network, see Definition [3.3.1][3] later on. In addition, for each node $A$ in $G$, we can look at the subgraph $G_A$ of cumulative parents of $A$. The interpretation $[[G_A]]$ is then also a state, namely the one that one obtains via state transformation. This has been described as ‘prediction’ in Section [1.9].

The above definition of Bayesian network is more general than usual, for instance because it allows the graph to contain joint states, like in (3.4), or discarding $\triangleright$, see also the discussion in [50].
Chapter 3. Directed Graphical Models

Exercises

3.1.1 For a state \( \omega \in D(X_1 \times X_2 \times X_3 \times X_4 \times X_5) \), describe its marginalisation \( \omega[1, 0, 1, 0, 0] \) as string diagram.

3.1.2 Check in detail how the two string diagrams in (3.3) are obtained by following the construction rules for string diagrams.

3.1.3 Consider the boxes in the wetness string diagram in (3.3) as a signature. An interpretation of the elements in this signature is described in Section [1.9] as states and channels \( wi, sp \ etc. \) Check that this interpretation extends to the whole wetness string diagram in (3.3) as a channel \( 1 \rightarrow D \times E \) given by:

\[
(wg \otimes sr) \circ (id \otimes \Delta) \circ (sp \otimes ra) \circ \Delta \circ wi.
\]

3.1.4 Extend the interpretation from Section [2.7] of the signature of the Asia string diagram in (3.3) to a similar interpretation of the whole Asia string diagram.

3.1.5 For \( A = \{a, b, c\} \) check that:

\[
\left\| A \right\| = \frac{1}{3}\left| a, a \right\rangle + \frac{1}{3}\left| b, b \right\rangle + \frac{1}{3}\left| c, c \right\rangle.
\]

3.1.6 Check that there are equalities of interpretations:

\[
\left\| A \times B \right\| = \left\| A \right\| \left\| B \right\| \quad \left\| A, B \right\| = \left\| A \times B \right\|.
\]

3.1.7 Verify that for each box, interpreted as a channel, one has:

\[
\left\| A \right\| = \left\| A \right\|.
\]

3.1.8 Recall Exercise [2.4.4] and check that the interpretation of

\[
\begin{array}{c}
  \circlearrowright \circlearrowleft \\
  g \\
  f
\end{array}
\]

is \( \left\| f \right\| \vdash \left\| g \right\| \).

Similarly, check that the interpretation of

\[
\begin{array}{c}
  \circlearrowleft \\
  h \\
  g \\
  f
\end{array}
\]

is \( \left\| g \right\| \gg \left\| f \right\| \vdash \left\| h \right\| = \left\| f \right\| \vdash \left\| g \right\| \ll \left\| h \right\| \).
3.2. Equations for string diagrams

see Proposition 2.4.3

3.1.9 Notice that the difference between $\omega \vdash p_1 \otimes p_2$ and $\sigma \vdash q_1 \& q_2$ can be clarified graphically as:

\[
\begin{array}{c}
\begin{array}{c}
\text{versus}
\end{array}
\end{array}
\]

See also Exercise 2.4.5

3.2 Equations for string diagrams

This section introduces equations $S_1 = S_2$ between string diagrams $S_1, S_2 \in SD(\Sigma)$ over the same signature $\Sigma$. These equation are ‘sound’, in the sense that they are respected by the string diagram interpretation of Subsection 3.1.2: $S_1 = S_2$ implies $\llbracket S_1 \rrbracket = \llbracket S_2 \rrbracket$. From now on we simplify the writing of string diagrams in the following way.

Convention 3.2.1. 1 We will write the arrows $\uparrow$ on the wires of string diagrams simply as $|$ and assume that the flow is always upwards in a string diagram, from bottom to top. Thus, even though we are not writing edges as arrows, we still consider string diagrams as directed graphs.

2 We drop the types of wires when they are clear from the context. Thus, wires in string diagrams are always typed, but we do not always write these types explicitly.

3 We become a bit sloppy about writing the interpretation function $\llbracket - \rrbracket$ for string diagrams explicitly. Sometimes we’ll say “the interpretation of $S$” instead of just $\llbracket S \rrbracket$, but also we sometimes simply write a string diagram, where we mean its interpretation as a channel. This can be confusing at first, so we will make it explicit when we start blurring the distinction between syntax and semantics.

Shift equations

String diagrams allow quite a bit of topological freedom, for instance, in the sense that parallel boxes may be shifted up and down, as expressed by the
following equations.

![Diagram of equations]

Semantically, this is justified by the equation:

\[
(\text{id} \otimes f) \circ (f \otimes \text{id}) = (f \otimes \text{id}) \circ (\text{id} \otimes f) = (\text{id} \otimes g) \circ (g \otimes \text{id}).
\]

Similarly, boxes may be pulled through swaps:

![Diagram of swaps]

These four equations do not only hold for boxes, but also for \(\uparrow\) and \(\downarrow\), since we consider them simply as boxes but with a special notation.

**Equations for wires**

Wires can be stretched arbitrarily in length, and successive swaps cancel each other out:

![Diagram of wires]

Wires of product type correspond to parallel wires, and wires of the empty product type \(1\) do nothing — as represented by the empty dashed box.

\[
A \times B \quad = \quad A \quad B \quad 1 \quad = \quad \text{[Empty dashed box]}
\]

**Equations for copying**

The copy string diagram \(\gamma\) also satisfies a number of equations, expressing associativity, commutativity and a co-unit property. Abstractly this makes \(\gamma\) a comonoid. This can be expressed graphically by:

![Diagram of copy equations]

\[
(3.5)
\]
Because copying is associative, it does not matter which order of copying we use in successive copying. Therefore we feel free to write such $n$-ary copying as coming from a single node $\bullet$, in:

\[
\Delta_n := \begin{array}{c}
\text{\ldots} \\
\end{array}
\quad \begin{array}{c}
\text{\ldots} \\
\end{array}
\]

(3.6)

Copying of products $A \times B$ and of the empty product $1$ comes with the following equations.

\[
A \times B = \begin{array}{c}
A \\
\text{\ldots} \\
\end{array} \begin{array}{c}
B \\
\text{\ldots} \\
\end{array}
\]

\[
1 = \begin{array}{c}
\text{\ldots} \\
\end{array}
\]

We should be careful that boxes can in general not be “pulled through” copiers, as expressed on the left below (see also Exercise 1.8.8).

\[
\text{\ldots} \\
\text{\ldots}
\]

but

\[
\begin{array}{c}
a \\
\end{array} = \begin{array}{c}
a \\
\end{array} \begin{array}{c}
a \\
\end{array}
\]

An exception to this is when copying follows a point state $\{|a\rangle\}$ for $a \in A$, as above, on the right.

**Equations for discarding and for uniform and point states**

The most important discarding equation is the one immediately below, saying that discarding after a box/channel is the same as discarding of the input altogether. We refer to this as: boxes are *unital*.

\[
\begin{array}{c}
\text{\ldots} \\
\end{array} \begin{array}{c}
A_1 \\
\text{\ldots} \\
\end{array} \begin{array}{c}
A_n \\
\text{\ldots} \\
\end{array} = \text{\ldots} \\
\text{\ldots}
\]

in particular

\[
\begin{array}{c}
\text{\ldots} \\
\end{array} = \text{\ldots}
\]

Discarding only some (not all) of the outgoing wires amounts to marginalisation.

Discarding a uniform state also erases everything. This may be seen as a special case of the above equation on the right, but still we like to formulate it explicitly, below on the left. The other equations deal with trivial cases.

\[
\begin{array}{c}
A \downarrow \\
\end{array} = \begin{array}{c}
\text{\ldots} \\
\end{array}
\]

\[
\begin{array}{c}
1 \downarrow = \begin{array}{c}
\text{\ldots} \\
\end{array}
\]

\[
\text{\ldots} = \text{\ldots}
\]

\[
\downarrow = \begin{array}{c}
\text{\ldots} \\
\end{array}
\]

\footnote{In a quantum setting one uses ‘causal’ instead of ‘unital’, see e.g. [16].}

143
Chapter 3. Directed Graphical Models

For product wires we have the following discard equations.

\[ A \times B = \tau = \tau = A \uparrow B \]

Similarly we have the following uniform state equations.

\[ A \times B = \tau = \tau = A \uparrow B \]

For point states we have for elements \( a \in A \) and \( b \in B \),

\[ (a, b) = a \uparrow b \]

Equations for logical operations

As is well-known, conjunction \& is associative and commutative:

\[ \& = \& = \& = \& \]

Moreover, conjunction has truth and falsity as unit and zero elements:

\[ \& = \& = 0 \]

Orthosupplement \( (-)^\perp \) is idempotent and turns 0 to 1 (and vice versa):

\[ \perp = \perp = 0 \]

Finally we look at the rules for scalar multiplication for predicates:

\[ r \cdot s = r \uparrow s = 1 = 0 = 0 = r \uparrow \& \]

144
3.2. Equations for string diagrams

These equations be used for diagrammatic equational reasoning. For instance we can now derive a variation on the last equation, where scalar multiplication on the second input of conjunction can be done after the conjunction:

\[
\begin{align*}
\& r &= \& r \quad \& r \quad \& r &\quad \& r \\
\end{align*}
\]

All of the above equations between string diagrams are sound, in the sense that they hold under all interpretations of string diagrams — as described at the end of the previous section. There are also completeness results in this area, but they require a deeper categorical analysis. For a systematic account of string diagrams we refer to the overview paper [90]. In this book we use string diagrams as convenient, intuitive notation with a precise semantics.

Exercises

3.2.1 Give a concrete description of the \( n \)-ary copy channel \( \Delta_n : A \to A^n \) in (3.6).

3.2.2 Prove by diagrammatic equational reasoning that:

\[
\begin{align*}
\& r \cdot s &= \& r \quad \& r, s &\quad \& rs
\end{align*}
\]

3.2.3 Consider for \( r \in [0, 1] \) the convex combination channel \( cc(r) : A \times A \to A \) given by \( cc(r)(a, a') = r|a| + (1 - r)|a'| \). It can be used as interpretation of an additional convex combination box:

Prove that it satisfies the following ‘barycentric’ equations, due to [96].

\[
\begin{align*}
1 &= r \\
1 - r &= s \\
1 - r + s &= 1 - r + s
\end{align*}
\]
3.3 Accessibility and joint states

Intuitively, a string diagram is called accessible if all its internal connections are also accessible externally. This property will be defined formally below, but it is best illustrated via an example. Below on the left we see the wetness Bayesian network as string diagram, like in (3.3) — but with lines instead of arrows. On the right is an accessible version of this string diagram, with (single) external connections to all its internal wires.

We are interested in such accessible versions because they lead in an easy way to joint states: the joint state for the wetness Bayesian network is the interpretation of the above accessible string diagram on the right, giving a distribution on the product set \( A \times B \times D \times C \times E \). Such joint distributions are conceptually important, but their size quickly grows out of hand. For instance, the joint distribution \( \omega \) for the wetness network is (with zero-probability terms omitted):

\[
\begin{align*}
0.06384[a, b, d, c, e] &+ 0.02736[a, b, d, c, e] + 0.0216[a, b, d, c', e'] \\
+ 0.00336[a, b, d', c, e] &+ 0.00144[a, b, d', c, e] + 0.0024[a, b, d', c', e'] \\
+ 0.215[a, b', d, c, e] &+ 0.09216[a, b', d, c, e] + 0.05376[a, b', d, c', e] \\
+ 0.02304[a, b', d', c, e'] &+ 0.096[a, b', d', c', e'] + 0.01995[a', b, d, c, e] \\
+ 0.00855[a', b, d, c, e'] &+ 0.243[a', b, d, c', e'] + 0.00105[a', b, d', c', e'] \\
+ 0.00045[a', b', d', c, e'] &+ 0.027[a', b, d, c', e'] + 0.0056[a', b', d, c, e] \\
+ 0.0024[a', b', d', c, e'] &+ 0.0014[a', b', d', c, e] + 0.0006[a', b', d', c, e'] \\
+ 0.09[a', b', d', c', e'].
\end{align*}
\]

This distribution is obtained as outcome of the interpreted accessible graph.
in (3.7):
\[ \omega = (\text{id} \otimes \text{id} \otimes \text{wg} \otimes \text{id} \otimes \text{sr}) \circ (\text{id} \otimes \Delta_2 \otimes \Delta_3) \circ (\text{id} \otimes \text{sp} \otimes \text{sr}) \circ \Delta_3 \circ \text{wi}. \]

Recall that \( \Delta_n \) is written for the \( n \)-ary copy channel, see (3.6).

Later on in this section we use our new insights into accessible string diagrams to re-examine the relation between crossover inference on joint states and channel-based inference, see Corollary 2.5.9. But we start with a more precise description.

**Definition 3.3.1.** 1 An **accessible string diagram** is defined inductively via the string diagram definitions steps (1) – (3) from Section 3.1, with sequential composition diagram (3.2) in step (4) extended with copiers on the internal wires:

\[
\begin{array}{c}
\cdots \\
S_1 \\
\cdots \\
\cdots \\
S_2 \\
\cdots \\
\end{array}
\]

In this way each ‘internal’ wire between string diagrams \( S_1 \) and \( S_2 \) is ‘externally’ accessible.

2 Each string diagram \( S \) can be made accessible by replacing each occurrence (3.2) of step (4) in its construction with the above modified step (3.8). We shall write \( \overline{S} \) for a choice of accessible version of \( S \).

3 A **joint state** or **joint distribution** associated with a Bayesian network \( G \in SD(\Sigma) \) is the interpretation \( \llbracket G \rrbracket \) of an accessible version \( \overline{G} \in SD(\Sigma) \) of the string diagram \( G \). It re-uses \( G \)'s interpretation of its boxes.

There are several subtle points related to the last two points.

**Remark 3.3.2.** 1 The above definition of accessible string diagram allows that there are multiple external connections to an internal wire, for instance if one of the internal wires between \( S_1 \) and \( S_2 \) in (3.8) is made accessible inside \( S_2 \), for instance because it is one of the outgoing wires of \( S_2 \). Such multiple accessibility is unproblematic, but in most cases we are happy with accessibility via a single wire, as in (3.7) on the right. Anyway, we can always purge unnecessary outgoing wires via discard \( \ddagger \).
2 The order of the outgoing wires in an accessible graph involves some level of arbitrariness. For instance, in the accessible wetness graph in (3.7) one could also imagine putting $C$ to the left of $D$ at the top of the diagram. This would involve some additional crossing of wires. The result would be essentially the same. The resulting joint state, obtained by interpretation, would then be in $\mathcal{D}(A \times B \times C \times D \times E)$. These differences are immaterial, but do require some care if one performs marginalisation.

Thus, strictly speaking, the accessible version $\mathcal{S}$ of a string diagram $S$ is determined only up to permutation of outgoing wires. In concrete cases we try to write accessible string diagrams in such a way that the number of crossings of wires is minimal.

With these points in mind we allow ourselves to speak about the joint state associated with a Bayesian network, in which duplicate outgoing wires are deleted. This joint state is determined up-to permutation of outgoing wires.

In the crossover inference corollary 2.5.9 we have seen how component-wise updates on a joint state correspond to channel-based inference. This theorem involved a binary state and only one channel. Now that we have a better handle on joint states via string diagrams, we can extend this correspondence to $n$-ary joint states and multiple channels. This is a theme that will be pursued further in this chapter. At this stage we only illustrate how this works for the wetness Bayesian network.

**Example 3.3.3.** In Exercise 2.7.1 we have formulated two inference questions for the wetness Bayesian network, namely:

1. What is the updated sprinkler distribution, given evidence of a slippery road? Using channel-based inference it is computed as:
   \[
   sp \gg \left(\frac{wi}{ra \leftarrow (sr \ll 1)}\right) = \frac{63}{560}b + \frac{197}{560}b^\perp.
   \]

2. What is the updated wet grass distribution, given evidence of a slippery road? It’s:
   \[
   wg \gg \left(\left((sp \otimes ra) \gg (\Delta \gg wi)\right)\right)_{\left(E \otimes (sr \ll 1)\right)} = \frac{4349}{5200}d + \frac{851}{5200}d^\perp.
   \]

Via (multiple applications) of crossover inference, see Corollary 2.5.9 these same updated probabilities can be obtained via the joint state $\omega \in \mathcal{D}(A \times B \times D \times C \times E)$ of the wetness Bayesian network, as mentioned at the beginning of this section.

Concretely, this works as follows. In the above first case we have (point) evidence $1_e: E \rightarrow [0, 1]$ on the set $E = \{e, e^\perp\}$. It is weakened (extended) to evidence $1 \otimes 1 \otimes 1 \otimes 1$ on the product $A \times B \times D \times C \times E$. Then it can be used to
update the joint state $\omega$. The required outcome then appears by marginalising on the sprinkler set $B$ in second position, as in:

$$\omega_{1\otimes 1\otimes 1\otimes 1\otimes 1}[0, 1, 0, 0, 0] = \frac{61}{260}|b\rangle + \frac{192}{260}|b^\perp\rangle$$

$$\approx 0.2423|b\rangle + 0.7577|b^\perp\rangle.$$ 

For the second inference question we use the same evidence, but we now marginalise on the wet grass set $D$, in third position:

$$\omega_{1\otimes 1\otimes 1\otimes 1\otimes 1}[0, 0, 1, 0, 0] = \frac{4349}{5200}|d\rangle + \frac{851}{5200}|d^\perp\rangle$$

$$\approx 0.8363|d\rangle + 0.1637|d^\perp\rangle.$$ 

We see that crossover inference is easier to express than channel-based inference, since we do not have to form the more complicated expressions with $\gg$ and $\ll$ in the above two points. Instead, we just have to update and marginalise at the right positions. Thus, it is easier from a mathematical perspective. But from a computational perspective crossover inference is more complicated, since these joint states grow exponentially in size (in the number of nodes in the graph). This topic will be continued in Section 3.10.

**Exercises**

3.3.1 Draw an accessible version of the Asia string diagram in (3.3). Write also the corresponding composition of channels that produces the associated joint state.
3.4 Hidden Markov models

Let’s start with a graphical description of an example of what is called a hidden Markov model:

This model has three ‘hidden’ elements, namely Cloudy, Sunny, and Rainy, representing the weather condition on a particular day. There are ‘temporal’ transitions with associated probabilities between these elements, as indicated by the labeled arrows. For instance, if it is cloudy today, then there is a 50% chance that it will be cloudy again tomorrow. There are also two ‘visible’ elements on the right: Stay-in, and Stay-out, describing two possible actions of a person, depending on the weather condition. There are transitions with probabilities from the hidden to the visible elements. The idea is that with every time step a transition is made between hidden elements, resulting in a visible outcome. Such steps may be repeated for a finite number of times — or even forever. The interaction between what is hidden and what can be observed is a key element of hidden Markov models. For instance, one may ask: given a certain initial state, how likely is it to see a consecutive sequence of the four visible elements: Stay-in, Stay-in, Go-out, Stay-in?

Hidden Markov models are simple statistical models that have many applications in temporal pattern recognition, in speech, handwriting or gestures, but also in robotics and in biological sequences. This section will briefly look into hidden Markov models, using the notation and terminology of channels. Indeed, a hidden Markov model can be defined easily in terms of channels and forward and backward transformation of states and observables. In addition, conditioning of states by observables can be used to formulate and answer elementary questions about hidden Markov models. Learning for Markov models will be described separately in Sections 4.5 and 4.6.
### 3.4. Hidden Markov models

**Definition 3.4.1.** A Markov model (or a Markov chain) is given by a set $X$ of 'internal positions', typically finite, with a 'transition' channel $t: X \rightarrow X$ and an initial state/distribution $\sigma \in D(X)$.

A hidden Markov model, often abbreviated as HMM, is a Markov model, as just described, with an additional 'emission' channel $e: X \rightarrow Y$, where $Y$ is a set of 'outputs'.

In the above illustration (3.9) we have as sets of positions and outputs:

- $X = \{\text{Cloudy}, \text{Sunny}, \text{Rainy}\}$
- $Y = \{\text{Stay-in}, \text{Go-out}\}$

with transition channel $t: X \rightarrow X$,

- $t(\text{Cloudy}) = 0.5 \{\text{Cloudy}\} + 0.2 \{\text{Sunny}\} + 0.3 \{\text{Rainy}\}$
- $t(\text{Sunny}) = 0.15 \{\text{Cloudy}\} + 0.8 \{\text{Sunny}\} + 0.05 \{\text{Rainy}\}$
- $t(\text{Rainy}) = 0.2 \{\text{Cloudy}\} + 0.2 \{\text{Sunny}\} + 0.6 \{\text{Rainy}\}$

and emission channel $e: X \rightarrow Y$,

- $e(\text{Cloudy}) = 0.5 \{\text{Stay-in}\} + 0.5 \{\text{Go-out}\}$
- $e(\text{Sunny}) = 0.2 \{\text{Stay-in}\} + 0.8 \{\text{Go-out}\}$
- $e(\text{Rainy}) = 0.9 \{\text{Stay-in}\} + 0.1 \{\text{Go-out}\}$

An initial state is missing in the picture (3.9).

In the literature on Markov models, the elements of the set $X$ are often called states. This clashes with the terminology in this book, since we use 'state' as synonym for 'distribution'. So, here we call $\sigma \in D(X)$ an (initial) state, and we call elements of $X$ (internal) positions. At the same time we may call $X$ the sample space. The transition channel $t: X \rightarrow X$ is an endo-channel on $X$, that is a channel from $X$ to itself. As a function, it is of the form $t: X \rightarrow D(X)$; it is an instance of a coalgebra, that is, a map of the form $A \rightarrow F(A)$ for a functor $F$, see [93] or [42] for more information.

In a Markov chain/model one can iteratively compute successor states. For an initial state $\sigma$ and transition channel $t$ one can form successor states via state transformation:

- $\sigma$
- $t \gg \sigma$
- $(t \gg \sigma) \gg (t \gg \sigma) = (t \gg t) \gg \sigma$ where $t^n = \begin{cases} \text{unit} & \text{if } n = 0 \\ t \circ t^{n-1} & \text{if } n > 0. \end{cases}$
- $t^n \gg \sigma$

In these transitions the state at stage $n + 1$ only depends on the state at stage $n$: 151
in order to predict a future step, all we need is the immediate predecessor state. This makes HMMs relatively easy dynamical models. Multi-stage dependencies can be handled as well, by enlarging the sample space, see Exercise 3.4.6 below.

One interesting problem in the area of Markov chains is to find a ‘stationary’ state $\sigma_\infty$ with $t \gg \sigma_\infty = \sigma_\infty$, see Exercise 3.4.2.

Here we are more interested in hidden Markov models $1 \xrightarrow{\sigma} X \xrightarrow{t} X \xrightarrow{e} Y$. The elements of the set $Y$ are observable — and hence sometimes called signals — whereas the elements of $X$ are hidden. Thus, many questions related to hidden Markov models concentrate on what one can learn about $X$ via $Y$, in a finite number of steps. Hidden Markov models are examples of models with latent variables.

We briefly discuss some basic issues related to HMMs in separate subsections. A recurring theme is the relationship between constructions involving a joint state and sequential constructions.

### 3.4.1 Validity in hidden Markov models

The first question that we like to address is: given a sequence of observables, what is their probability (validity) in a HMM? Standardly in the literature, one only looks at the probability of a sequence of point observations (elements), but here we use a more general approach. After all, one may not be certain about observing a specific point at a particular state, or some point observations may be missing; in the latter case one may wish to replace them by a constant (uniform) observation.

We proceed by defining validity of the sequence of observables in a joint state first; subsequently we look at (standard) algorithms for computing these validities more efficiently. We thus start by defining a relevant joint state.

We fix a HMM $1 \xrightarrow{\sigma} X \xrightarrow{t} X \xrightarrow{e} Y$. For each $n \in \mathbb{N}$ a channel $\langle e, t \rangle_n: X \rightarrow Y^n \times X$ is defined in the following manner:

$$
\langle e, t \rangle_0 := \left( \xrightarrow{id} X \cong I \times X \cong Y^0 \times X \right)
$$

$$
\langle e, t \rangle_{n+1} := \left( \xrightarrow{id \otimes (e,t)} Y^n \times X \xrightarrow{id \otimes (e,t)} Y^n \times (Y \times X) \cong Y^{n+1} \times X \right)
$$

We recall that the tuple $\langle e, t \rangle$ of channels is $(e \otimes t) \in \Delta$, see Definition 3.8.10. With these tuples we can form a joint state $\langle e, t \rangle_n \gg \sigma \in \mathcal{D}(Y^n \times X)$. As an
3.4. Hidden Markov models

We consider the combined likelihood of a sequence of observables on the set \( Y \) in a hidden Markov model. In the literature these observables are typically point predicates \( \mathbf{1}_y : Y \to [0, 1] \), for \( y \in Y \), but, as mentioned, here we allow more general observables \( Y \to \mathbb{R} \).

**Definition 3.4.2.** Let \( \mathcal{H} = (1 \xrightarrow{\sigma} X \xrightarrow{t} X \xrightarrow{e} Y) \) be a hidden Markov model and let \( \vec{p} = p_1, \ldots, p_n \) be a list of observables on \( Y \). The validity \( \mathcal{H} \models \vec{p} \) of this sequence \( \vec{p} \) in the model \( \mathcal{H} \) is defined via the tuples (3.10) as:

\[
\mathcal{H} \models \vec{p} \triangleq \langle e, t \rangle_n \gg \sigma \models [1, \ldots, 1, 0] = p_1 \otimes \cdots \otimes p_n
\]

\[
\mathcal{H} \models \vec{p} \triangleq \langle e, t \rangle_n \gg \sigma \models p_1 \otimes \cdots \otimes p_n \otimes \mathbf{1}
\]

\[
\mathcal{H} \models \vec{p} \triangleq \langle e, t \rangle_n \ll (p_1 \otimes \cdots \otimes p_n \otimes \mathbf{1})
\]

The marginalisation mask \( [1, \ldots, 1, 0] \) contains \( n \) times the number 1. It ensures that the \( X \) outcome in (3.11) is discarded.

We describe an alternative way to formulate this validity without using the (big) joint state on \( Y^n \times X \). It forms the essence of the classical ‘forward’ and ‘backward’ algorithms for validity in HMMs, see e.g. [87] or [58, App. A]. An alternative algorithm is described in Exercise 3.4.4.

**Proposition 3.4.3.** The HMM-validity (3.12) can be computed as:

\[
\mathcal{H} \models \vec{p} = \sigma \models (e \ll p_1) \&
\]

\[
t \ll ((e \ll p_2) \& \cdots
\]

\[
t \ll (e \ll p_3) \& \cdots
\]

\[
t \ll (e \ll p_n) \cdots )
\]

This validity can be calculated recursively in forward manner as:

\[
\sigma \models \alpha(\vec{p}) \quad \text{where} \quad \left\{
\begin{array}{ll}
\alpha([q]) = e \ll q \\
\alpha([q]++\vec{q}) = (e \ll q) \& (t \ll \alpha(\vec{q})).
\end{array}
\right.
\]
Alternatively, this validity can be calculated recursively in backward manner as:

\[ \sigma \models \beta(\vec{q}, 1) \] where

\[ \beta([q]) = q \]

\[ \beta(\vec{q} ++ [q_n, q_{n+1}]) = \beta(\vec{q} ++ [(e \ll q_n) \& (t \ll q_{n+1})]) \]

**Proof.** We first prove, by induction on \( n \geq 1 \) that for observables \( p_i \) on \( Y \) and \( q \) on \( X \) one has:

\[ \langle e, t \rangle_n \ll (p_1 \otimes \cdots \otimes p_n \otimes q) \]

\[ = (e \ll p_1) \& t \ll ((e \ll p_2) \& t \ll (\cdots t \ll ((e \ll p_n) \& t \ll q) \cdots)) \] (\( * \))

The base case \( n = 1 \) is easy:

\[ \langle e, t \rangle_1 \ll (p_1 \otimes q) = \Delta \ll ((e \otimes t) \ll (p_1 \otimes q)) \]

\[ = \Delta \ll ((e \ll p_1) \otimes (t \ll q)) \]

\[ = (e \ll p_1) \& (t \ll q) \]

by Exercise 2.4.6

\[ \text{by Exercise 2.4.5} \]

For the induction step we reason as follows.

\[ \langle e, t \rangle_{n+1} \ll (p_1 \otimes \cdots \otimes p_n \otimes p_{n+1} \otimes q) \]

\[ = \langle e, t \rangle_n \ll ((\text{id}^n \otimes (t, e)) \ll (p_1 \otimes \cdots \otimes p_n \otimes p_{n+1} \otimes q)) \]

\[ = \langle e, t \rangle_n \ll (p_1 \otimes \cdots \otimes p_n \otimes ((e, t) \ll (p_{n+1} \otimes q))) \]

\[ = \langle e, t \rangle_n \ll (p_1 \otimes \cdots \otimes p_n \otimes ((e \ll p_{n+1}) \& (t \ll q))) \]

\[ = (e \ll p_1) \& t \ll ((e \ll p_2) \& t \ll (\cdots t \ll ((e \ll p_n) \& t \ll q) \cdots)) \]

\[ \text{as just shown} \]

\[ \text{we can now prove Equation (3.13):} \]

\[ \mathcal{H} \models \overline{p} \]

\[ \sigma \models \langle e, t \rangle_n \ll (p_1 \otimes \cdots \otimes p_n \otimes 1) \]

\[ \models \]

\[ (e \ll p_1) \& t \ll ((e \ll p_2) \& t \ll (\cdots t \ll ((e \ll p_n) \& t \ll 1) \cdots)) \]

\[ = (e \ll p_1) \& t \ll ((e \ll p_2) \& t \ll (\cdots t \ll (e \ll p_n) \cdots)). \]

\[ \square \]

### 3.4.2 Filtering

Given a sequence \( \overline{p} \) of factors, one can compute their validity \( \mathcal{H} \models \overline{p} \) in a HMM \( \mathcal{H} \), as described above. But we can also use these factors to ‘guide’ the evolution of the HMM. At each state \( i \) the factor \( p_i \) is used to update the current state, via backward inference. The new state is then moved forward via the transition function. This process is called filtering, after the Kalman filter from the 1960s that is used for instance in trajectory optimisation in navigation and in rocket control (e.g. for the Apollo program). The system can evolve
automatically via its transition function, but observations at regular intervals can update (correct) the current state.

**Definition 3.4.4.** Let \( \mathcal{H} = (1 \xrightarrow{\sigma} X \xrightarrow{\tau} X \xrightarrow{\tau} Y) \) be a hidden Markov model and let \( \bar{p} = p_1, \ldots, p_n \) be a list of factors on \( Y \). It gives rise to the filtered sequence of states \( \sigma_1, \sigma_1, \ldots, \sigma_{n+1} \in \mathcal{D}(X) \) following the observe-update-proceed principle:

\[
\sigma_1 \coloneqq \sigma \quad \text{and} \quad \sigma_{i+1} \coloneqq t \gg (\sigma_{i+1} \mid e \ll p_i).
\]

In the terminology of Definition 2.5.1, the definition of the state \( \sigma_{i+1} \) involves both forward and backward inference. Below we show that the final state \( \sigma_{n+1} \) in the filtered sequence can also be obtained via crossover inference on a joint state, obtained via the tuple channels (3.10). This fact gives a theoretical justification, but is of little practical relevance — since joint states quickly become too big to handle.

**Proposition 3.4.5.** In the context of Definition 3.4.4

\[
\sigma_{n+1} = (\langle e, t \rangle_n \gg \sigma) \mid_{p_1 \otimes \cdots \otimes p_n \otimes 1}[0, \ldots, 0, 1]
\]

*The marginalisation mask \([0, \ldots, 0, 1]\) has \( n \) zero’s.

**Proof.** By induction on \( n \geq 1 \). The base case with \( \langle e, t \rangle_1 = \langle e, t \rangle \) is handled as follows.

\[
\begin{align*}
(\langle e, t \rangle \gg \sigma) \mid_{p_1 \otimes 1}[0, 1] \\
&= \pi_2 \gg (\langle e \mid p_1, t \gg (\sigma_{\langle e, t \rangle \mid e \ll (p_1 \otimes 1)}) \quad \text{by Corollary 2.5.8(2)} \\
&= t \gg (\sigma_{\langle e \mid p_1}) \\
&= \sigma_2.
\end{align*}
\]
The last equation uses Exercise 2.2.4 and Definition 3.4.2.

Proof

We briefly consider one more question for HMMs: given a sequence of observations, Bayes’ rule for HMMs — or more accurately, the product rule for HMMs, see Chapter 3. Directed Graphical Models.

The next consequence of the previous proposition may be understood as Bayes’ rule for HMMs — or more accurately, the product rule for HMMs, see Proposition 2.3.3.

Corollary 3.4.6. Still in the context of Definition 3.4.4 let q be a predicate on X. Its validity in the final state in the filtered sequence is given by:

\[ \sigma_{n+1} = (e, t)_{n+1} \Rightarrow \sigma \equiv \sigma_{1} \otimes \cdots \otimes \sigma_{n} \otimes q. \]

Proof. This follows from Bayes’ rule, in Proposition 2.3.3(1):

\[ \sigma_{n+1} \equiv q = (e, t)_{n+1} \Rightarrow \sigma \equiv (e, t)_{n} \Rightarrow \sigma_{1} \otimes \cdots \otimes \sigma_{n} \otimes q. \]

The last equation uses Exercise 2.2.4 and Definition 3.4.2.

3.4.3 Finding the most likely sequence of hidden elements

We briefly consider one more question for HMMs: given a sequence of observations, what is the most likely path of internal positions that produces these
observations? This is also known as the decoding problem. It explicitly asks for the most likely path, and not for the most likely individual position at each stage. Thus it involves taking the argmax of a joint state. We concentrate on giving a definition of the solution, and only sketch how to obtain it efficiently.

We start by constructing the state below, as interpreted string diagram, for a given HMM $\sigma \rightarrow X \rightarrow t \rightarrow X$.

![State Diagram](image)

These maps $⟨\text{id}, t, e⟩_n$ are obtained by induction on $n$:

$$
⟨\text{id}, t, e⟩_0 := (X \xrightarrow{\text{id}} X \cong 1 \times 1 \times 1 \cong X^0 \times X \times Y^0)
$$

$$
⟨\text{id}, t, e⟩_{n+1} := (X \xrightarrow{\text{id}, t, e}_{n+1} X^n \times X \times Y^n \xrightarrow{\text{id}, t, e}_{n+1} X^n \times (X \times X \times Y) \times Y^n \xrightarrow{\text{id}, t, e}_{n+1} X^{n+1} \times X \times Y^{n+1})
$$

**Definition 3.4.7.** Let $1 \xrightarrow{\sigma} X \xrightarrow{t} X \xrightarrow{e} Y$ be a HMM and let $\vec{p} = p_1, \ldots, p_n$ be a sequence of factors on $Y$. Given these factors as successive observations, the most likely path of elements of the sample space $X$, as an $n$-tuple in $X^n$, is obtained as:

$$
\arg\max\left(\left(⟨\text{id}, t, e⟩_n \gg \sigma\right)_{\text{id}, t, e}_{1, \ldots, p_n} [1, \ldots, 1, 0, 0, \ldots, 0]\right).
$$

This expression involves updating the joint state (3.14) with the factors $p_i$, in reversed order, and then taking its marginal so that the state with the first $n$ outcomes in $X$ remains. The argmax of the latter state gives the sequence in $X^n$ that we are after.

The expression (3.15) is important conceptually, but not computationally.
158  Chapter 3. Directed Graphical Models

It is inefficient to compute, first of all because it involves a joint state that grows exponentially in \(n\), and secondly because the conditioning involves normalisation that is irrelevant when we take the argmax.

We give an impression of what \(3.15\) amounts to for \(n = 3\). We first focus on the expression within argmax (\(\cdot\)). The marginalisation and conditioning amount produce an state on \(X \times X \times X\) of the form:

\[
\sum_{x,y,z} \left( \phi((\text{id}, t, e)_2 \gg \sigma)(x_1, x_2, x_3, y_1, y_2, y_3) \cdot p_1(y_3) \cdot p_2(y_2) \cdot p_1(y_1) \right) \\
\left( \text{id}, t, e \right)_1 / \sigma = \mathbf{1} \otimes \mathbf{1} \otimes \mathbf{1} \otimes p_3 \otimes p_2 \otimes p_1
\]

\[
\approx \sum_{x,y,z} \sigma(x_1) \cdot t(x_2)(x_3) \cdot t(x_3)(x) \\
\cdot e(x_1)(y_3) \cdot e(x_2)(y_2) \cdot e(x_3)(y_1) \cdot p_3(y_3) \cdot p_2(y_2) \cdot p_1(y_1)
\]

\[
= \sigma(x_1) \cdot \left( \sum_{y_3} e(x_1)(y_1) \cdot p_1(y_1) \right) \cdot t(x_1)(x_2) \cdot \left( \sum_{y_2} e(x_2)(y_2) \cdot p_2(y_2) \right)
\]

\[
\cdot t(x_2)(x_3) \cdot \left( \sum_{y_3} e(x_3)(y_3) \cdot p_3(y_3) \right) \cdot \left( \sum_{y_3} t(x_3)(x) \right)
\]

\[
= \sigma(x_1) \cdot (e \ll p_1)(x_1) \cdot t(x_1)(x_2) \cdot (e \ll p_2)(x_2) \cdot t(x_2)(x_3) \cdot (e \ll p_3)(x_3).
\]

The argmax of the latter expression can then be computed as in Exercise 3.4.7.

This approach is known as the Viterbi algorithm, see e.g. [87] [58].

Exercises

3.4.1 Consider the HMM example 3.9 with initial state \(\sigma = 1\{\text{Cloudy}\}\).

1. Compute successive states \(t^n \gg \sigma\) for \(n = 0, 1, 2, 3\).
2. Compute successive observations \(e \gg (t^n \gg \sigma)\) for \(n = 0, 1, 2, 3\).
3. Check that the validity of the sequence of (point-predicate) observations Go-out, Stay-in, Stay-in is 0.1674.
4. Show that filtering, as in Definition 3.4.4, with these same three (point) observations yields as final outcome:

\[
\frac{1867}{6595}\{\text{Cloudy}\} + \frac{347}{1195}\{\text{Sunny}\} + \frac{15817}{3480}\{\text{Rainy}\}
\]

\[
\approx 0.2788\{\text{Cloudy}\} + 0.2487\{\text{Sunny}\} + 0.4724\{\text{Rainy}\}.
\]

3.4.2 Consider the transition channel \(t\) associated with the HMM example 3.9. Check that in order to find a stationary state \(\sigma_\infty = x\{\text{Cloudy}\} + y\{\text{Sunny}\} + z\{\text{Rainy}\}\) one has to solve the equations:

\[
x = 0.5x + 0.15y + 0.2z
\]

\[
y = 0.2x + 0.8y + 0.2z
\]

\[
z = 0.3x + 0.05y + 0.6z
\]

Deduce that \(\sigma_\infty = 0.25\{\text{Cloudy}\} + 0.5\{\text{Sunny}\} + 0.25\{\text{Rainy}\}\) and double-check that \(t \gg \sigma_\infty = \sigma_\infty\).
3.4.3 (The set-up of this exercise is copied from Machine Learning lecture notes of Doina Precup.) Consider a 5-state hallway of the form:

```
| 1 | 2 | 3 | 4 | 5 |
```

Thus we use a space \( X = \{1, 2, 3, 4, 5\} \) of positions, together with a space \( Y = \{2, 3\} \) of outputs, for the number of surrounding walls. The transition and emission channels \( t : X \rightarrow X \) and \( e : X \rightarrow Y \) for a robot in this hallway are given by:

\[
\begin{align*}
    t(1) &= \frac{3}{4} |1\rangle + \frac{1}{4} |2\rangle \\
    t(2) &= \frac{1}{2} |1\rangle + \frac{1}{2} |2\rangle + \frac{1}{2} |3\rangle \\
    t(3) &= \frac{1}{4} |2\rangle + \frac{1}{2} |3\rangle + \frac{1}{2} |4\rangle \\
    t(4) &= \frac{1}{4} |3\rangle + \frac{1}{2} |3\rangle + \frac{1}{2} |5\rangle \\
    t(5) &= \frac{1}{4} |4\rangle + \frac{1}{2} |5\rangle \\
\end{align*}
\]

\[
\begin{align*}
    e(1) &= 1 |3\rangle \\
    e(2) &= 1 |2\rangle \\
    e(3) &= 1 |2\rangle \\
    e(4) &= 1 |2\rangle \\
    e(5) &= 1 |3\rangle \\
\end{align*}
\]

We use \( \sigma = 1 |3\rangle \) as start state, and we have a sequence of observations \( \alpha = [2, 2, 3, 2, 3, 3] \), formally as a sequence of point predicates \( \{1_2, 1_2, 1_3, 1_2, 1_3, 1_3\} \).

1. Check that \((\sigma, t, e) \models \alpha = \frac{3}{312}\).

2. Next we filter with the sequence \( \alpha \). Show that it leads successively to the following states \( \sigma_i \) as in Definition 3.4.4

\[
\begin{align*}
    \sigma_1 &:= \sigma = 1 |3\rangle \\
    \sigma_2 &= \frac{1}{4} |2\rangle + \frac{1}{2} |3\rangle + \frac{1}{4} |4\rangle \\
    \sigma_3 &= \frac{1}{5} |1\rangle + \frac{1}{4} |2\rangle + \frac{1}{2} |3\rangle + \frac{1}{4} |4\rangle + \frac{1}{10} |5\rangle \\
    \sigma_4 &= \frac{1}{8} |1\rangle + \frac{1}{4} |2\rangle + \frac{1}{2} |4\rangle + \frac{3}{8} |5\rangle \\
    \sigma_5 &= \frac{1}{8} |1\rangle + \frac{1}{4} |2\rangle + \frac{1}{2} |3\rangle + \frac{1}{4} |4\rangle + \frac{3}{8} |5\rangle \\
    \sigma_6 &= \frac{1}{8} |1\rangle + \frac{1}{4} |2\rangle + \frac{1}{4} |4\rangle + \frac{3}{8} |5\rangle \\
    \sigma_7 &= \frac{1}{8} |1\rangle + \frac{1}{4} |2\rangle + \frac{1}{4} |4\rangle + \frac{3}{8} |5\rangle \\
\end{align*}
\]

3. Show that a most likely sequence of states giving rise to the sequence of observations \( \alpha \) is \([3, 2, 1, 2, 1, 1] \). Is there an alternative?

3.4.4 Apply Bayes’ rule to the validity formulation (3.13) in order to prove the correctness of the following HMM validity algorithm.

\[
(\sigma, t, e) \models [] \models 1 \\
(\sigma, t, e) \models [p] \models \vec{p} := (\sigma \models e \ll p) \cdot (t \Rightarrow (\sigma|_{e<p}), t, e) \models \vec{p}.
\]

(Notice the connection with filtering from Definition 3.4.4)
3.4.5 A random walk is a Markov model \( d: \mathbb{Z} \rightarrow \mathbb{Z} \) given by \( d(n) = r(n - 1) + (1 - r)n + 1 \) for some \( r \in [0, 1] \). This captures the idea that a step-to-the-left or a step-to-the-right are the only possible transitions. (The letter ‘\( d \)’ hints at modeling a drunkard.)

1 Start from initial state \( \sigma = 1|0\rangle \in D(\mathbb{Z}) \) and describe a couple of subsequent states \( d \gg \sigma, \ d^2 \gg \sigma, \ d^3 \gg \sigma, \ldots \) Which pattern emerges?

2 Prove that for \( K \in \mathbb{N} \),

\[
d^K \gg \sigma = \sum_{0 \leq k \leq K} \binom{K}{k}(1 - r)^k r^{K-k} |2k - K\rangle
\]

3.4.6 A Markov chain \( X \rightarrow X \) has a ‘one-stage history’ only, in the sense that the state at stage \( n + 1 \) depends only on the state at stage \( n \). The following situation from [87, Chap. III, Ex. 4.4] involves a two-stage history.

Suppose that whether or not it rains today depends on weather conditions through the last two days. Specifically, suppose that if it has rained for the past two days, then it will rain tomorrow with probability 0.7; if it rained today but not yesterday, then it will rain tomorrow with probability 0.5; if it rained yesterday but not today, then it will rain tomorrow with probability 0.4; if it has not rained in the past two days, then it will rain tomorrow with probability 0.2.

1 Write \( R = \{r, r^\perp\} \) for the state space of rain and no-rain outcomes, and capture the above probabilities via a channel \( c: R \times R \rightarrow R \).

2 Turn this channel \( c \) into a Markov chain \( \langle \pi_2, c \rangle: R \times R \rightarrow R \times R \), where the second component of \( R \times R \) describes whether or not it rains on the current day, and the first component on the previous day. Describe \( \langle \pi_2, c \rangle \) both as a function and as a string diagram.

3 Generalise this approach to a history of length \( N > 1 \): turn a channel \( X^N \rightarrow X \) into a Markov model \( X^N \rightarrow X^N \), where the relevant history is incorporated into the sample space.

3.4.7 Use the approach of the previous exercise to turn a hidden Markov model into a Markov model.

3.4.8 Let \( \mathcal{H}_1 \) and \( \mathcal{H}_2 \) be two HMMs. Define their parallel product \( \mathcal{H}_1 \otimes \mathcal{H}_2 \) using the tensor operation \( \otimes \) on states and channels.
3.5 Disintegration

In Subsections 1.3.2 and 1.4.5 we have seen how a binary relation $R \in \mathcal{P}(A \times B)$ on $A \times B$ corresponds to a $\mathcal{P}$-channel $A \rightarrow \mathcal{P}(B)$, and similarly how a multiset $\psi \in \mathcal{M}(A \times B)$ corresponds to an $\mathcal{M}$-channel $A \rightarrow \mathcal{M}(B)$. This phenomenon was called: extraction of a channel from a joint state. This section describes the analogue for probabilistic binary/joint states and channels. It turns out to be more subtle because probabilistic extraction requires normalisation — in order to ensure that the unitality requirement of a $\mathcal{D}$-channel: multiplicities need to add up to one.

In the probabilistic case this extraction is also called disintegration. It corresponds to a well-known phenomenon, namely that one can write a joint probability $P(x, y)$ in terms of a conditional probability as:

$$P(x, y) = P(y \mid x) \cdot P(x).$$

The mapping $x \mapsto P(y \mid x)$ is the channel involved, and $P(x)$ refers to the first marginal of $P(x, y)$.

Our description of disintegration makes use of the graphical calculus that we introduced earlier in this chapter. The formalisation of disintegration in category theory is not there (yet), but the closely related concept of Bayesian inversion has a nice categorical description as a ‘dagger’, see Section 3.8 below.

We start by formulating the property of ‘full support’ in string diagrammatic terms. It is needed as pre-condition in disintegration.

**Definition 3.5.1.** Let $X$ be a non-empty finite set. We say that a distribution or multiset $\omega$ on $X$ has full support if $\omega(x) > 0$ for each $x \in X$.

More generally, we say that a $\mathcal{D}$- or $\mathcal{M}$-channel $c : Y \rightarrow X$ has full support if the distribution/multiset $c(y)$ has full support for each $y$. This thus means that $c(y)(x) > 0$ for all $y \in Y$ and $x \in X$.

**Lemma 3.5.2.** Let $\omega \in \mathcal{D}(X)$, where $X$ is non-empty and finite. The following three points are equivalent:

1. $\omega$ has full support;
2. there is a predicate $p$ on $X$ and probability $r \in [0, 1]$ such that $\omega(x) \cdot p(x) = r$ for each $x \in X$.
3. there is a predicate $p$ on $X$ such that $\omega|_p$ is the uniform distribution on $X$.

**Proof.** Suppose that the finite set $X$ has $N \geq 1$ elements. For the implication $[1] \Rightarrow [2]$, let $\omega \in \mathcal{D}(X)$ have full support, so that $\omega(x) > 0$ for each $x \in X$. 

161
Since $\omega(x) \leq 1$, we get $\frac{1}{\omega(x)} \geq 1$ so that $t \geq \sum_x \frac{1}{\omega(x)} \geq N$ and $t \cdot \omega(x) \geq 1$, for each $x \in X$. Take $r = \frac{1}{t} \leq \frac{1}{N} \leq 1$, and put $p(x) = \frac{1}{t \omega(x)}$. Then:

$$\omega(x) \cdot p(x) = \frac{\omega(x)}{t} = \frac{1}{r} = r.$$  

Next, for $(2) \Rightarrow (3)$, suppose $\omega(x) \cdot p(x) = r$, for some predicate $p$ and probability $r$. Then $\omega \models p = \sum_x \omega(x) \cdot p(x) = N \cdot r$. Hence:

$$\omega_p(x) = \frac{\omega(x) \cdot p(x)}{\omega \models p} = \frac{r}{N \cdot r} = \frac{1}{N} = v_X(x).$$

For the final step $(3) \Rightarrow (1)$, let $\omega_p = v_X$. This requires that $\omega \models p \neq 0$ and thus that $\omega(x) \cdot p(x) = \frac{\omega \models p}{N} > 0$ for each $x \in X$. But then $\omega(x) > 0$.

This result allows us to express the full support property in diagrammatic terms. This will be needed as precondition later on.

**Definition 3.5.3.** A state box $s$ has full support if there is a predicate box $p$ and a scalar box $r$, for some $r \in [0, 1]$, with an equation as on the left below.

Similarly, we say that a channel box $f$ has full support if there are predicates $p, q$ with an equation as above on the right.

Strictly speaking, we didn’t have to distinguish states and channels in this definition, since that state-case is a special instance of the channel-case, namely for $A = 1$. In the sequel we shall no longer make this distinction.

We now come to the main definition of this section. It involves turning a conditional probability $P(b, c \mid a)$ into $P(c \mid b, a)$ such that

$$P(b, c \mid a) = P(c \mid b, a) \cdot P(b \mid a).$$

This is captured diagrammatically in (3.16) below.

**Definition 3.5.4.** Consider a box $f$

for which $f[1, 0] = \parallel f$ has full support.
3.5. Disintegration

We say that $f$ admits disintegration if there is a unique ‘disintegration’ box $f'$ from $B, A$ to $C$ satisfying the equation:

$$f' = \begin{array}{c} \text{B} \\ f' \\ \text{C} \end{array} = \begin{array}{c} \text{B} \\ f \\ \text{C} \end{array}$$

(3.16)

Uniqueness of $f'$ in this definition means: for each box $g$ from $B, A$ to $C$ and for each box $h$ from $C$ to $A$ with full support one has:

$$g \cdot h = \begin{array}{c} \text{B} \\ f \\ \text{C} \end{array} = \begin{array}{c} \text{A} \\ f' \end{array} \Rightarrow h = f' \text{ and } g = f'$$

The first equation $h = f[1, 0]$ in the conclusion is obtained simply by applying discard $\oplus$ to two right-most outgoing wires in the assumption (and using that $g$ is unital):

$$h = \begin{array}{c} \text{B} \\ f \\ \text{A} \end{array} = \begin{array}{c} \text{B} \\ f' \end{array} \text{ and } \begin{array}{c} \text{C} \\ h \end{array} = \begin{array}{c} \text{C} \\ f \end{array}$$

The second equation $g = f'$ in the conclusion expresses uniqueness of the disintegration box $f'$.

We shall use the notation $\text{extr}_1(f)$ or $f[0, 1\mid 1, 0]$ for this disintegration box $f'$. This notation will be explained in greater generality below.

The first thing to do is to check that this newly introduced construction is sound.

**Proposition 3.5.5.** Disintegration as described in Definition 3.5.4 exists for probabilistic channels.
We then write
\[ \omega \] instead: lists containing as only elements 0 or 1.

Without the full support requirement, disintegrations may still exist, but they are not unique, see Example 3.6.1 (2) for an illustration.

Next we elaborate on the notation that we will use for disintegration.

**Remark 3.5.6.** In traditional notation in probability theory one simply omits variables to express marginalisation. For instance, for a distribution \( \omega \in \mathcal{D}(X_1 \times X_2 \times X_3 \times X_4) \), considered as function \( \omega(x_1, x_2, x_3, x_4) \) in four variables \( x_i \), one writes:

\[ \omega(x_2, x_3) \quad \text{for the marginal} \quad \sum_{x_1, x_4} \omega(x_1, x_2, x_3, x_4). \]

We have been using *masks* instead: lists containing as only elements 0 or 1. We then write \( \omega[0, 1, 1, 0] \in \mathcal{D}(X_2 \times X_3) \) to express the above marginalisation, with a 0 (resp. a 1) at position \( i \) in the mask meaning that the \( i \)-th variable in the distribution is discarded (resp. kept).

We like to use a similar notation mask-style for disintegration, mimicking the traditional notation \( \omega(x_1, x_4 \mid x_2, x_3) \). This requires two masks, separated

---

**Proof.** Let \( f: A \to B \times C \) be a channel such that \( f[1, 0]: A \to B \) has full support. The latter means that for each \( a \in A \) and \( b \in B \) one has \( f[1, 0](a)(b) = \sum_c f(a)(b, c) \neq 0 \). Then we can define \( f': B \times A \to C \) as:

\[
f'(b, a) = \sum_c \frac{f(a)(b, c)}{f[1, 0](a)(b)} | c). \tag{3.17}
\]

By the full support assumption this is well-defined.

The left-hand-side of Equation (3.16) evaluates as:

\[
\left((\id \otimes f') \circ (\Delta \otimes \id) \circ (f[1, 0] \otimes \id) \circ \Delta\right)(a)(b, c)
\]

\[= \sum_{x,y} \id(y)(b) \cdot f'(y, x)(c) \cdot f[1, 0](a)(y) \cdot \id(x)(a)
\]

\[= f'(b, a)(c) \cdot f[1, 0](a)(b)
\]

\[= f(a)(b, c).
\]

For uniqueness, assume \((\id \otimes g) \circ (\Delta \otimes \id) \circ (h \otimes \id) \circ \Delta = f\). As already mentioned, we can obtain \( h = f[1, 0] \) via diagrammatic reasoning. Here we reason element-wise. By assumption, \( g(b, a)(c) \cdot h(a)(b) = f(a)(b, c) \). This gives:

\[h(a)(b) = 1 \cdot h(a)(b) = (\sum_c g(b, a)(c)) \cdot h(a)(b) = \sum_c g(b, a)(c) \cdot h(a)(b)
\]

\[= \sum_c f(a)(b, c)
\]

\[= f[1, 0](a)(b).
\]

Hence \( h = f[1, 0] \). But then \( g = f' \) since:

\[g(b, a)(c) = \frac{f(a)(b, c)}{h(a)(b)} = \frac{f(a)(b, c)}{f[1, 0](a)(b)} = f'(b, a)(c). \square
\]
by the sign ‘|’ for conditional probability. We sketch how it works for a box $f$ from $A$ to $B_1, \ldots, B_n$ in a string diagram. The notation 

$$f[N \mid M]$$

will be used for a disintegration channel, in the following manner:

1. masks $M, N$ must both be of length $n$;
2. $M, N$ must be disjoint: there is no position $i$ with a 1 both in $M$ and in $N$;
3. the marginal $f[M]$ must have full support;
4. the domain of the disintegration box $f[N \mid M]$ is $\bar{B} \cap M, A$, where $\bar{B} \cap M$ contain precisely those $B_i$ with a 1 in $M$ at position $i$;
5. the codomain of $f[N \mid M]$ is $\bar{B} \cap N$.

$f[N \mid M]$ is unique in satisfying an “obvious” adaptation of Equation (3.16), in which $f[M \cup N]$ is equal to a string diagram consisting of $f[M]$ suitably followed by $f[N \mid M]$.

How this really works is best illustrated via a concrete example. Consider the box $f$ on the left below, with five output wires. We elaborate the disintegration $f[1, 0, 0, 0, 1 \mid 0, 1, 0, 1, 0]$, as on the right.

The disintegration box $f[1, 0, 0, 0, 1 \mid 0, 1, 0, 1, 0]$ is unique in satisfying:

$$f[1, 0, 0, 0, 1 \mid 0, 1, 0, 1, 0] = f$$

In traditional notation one could express this equation as:

$$f(b_1, b_5 \mid b_2, b_4, a) \cdot f(b_2, b_4 \mid a) = f(b_1, b_2, b_4, b_5 \mid a).$$

Two points are still worth noticing.
• It is not required that at position $i$ there is a 1 either in mask $M$ or in mask $N$ when we form $f[N \mid M]$. If there is a 0 at $i$ both in $M$ and in $N$, then the wire at $i$ is discarded altogether. This happens in the above illustration for the third wire: the string diagram on the right-hand-side of the equation is $f[M \cup N] = f[1, 1, 0, 1, 1]$.

• The above disintegration $f[1, 0, 0, 0, 1 \mid 0, 1, 0, 1, 0]$ can be obtained from the ‘simple’, one-wire version of disintegration in Definition 3.5.4 by first suitably rearranging wires and combining them via products. How to do this precisely is left as an exercise (see below).

Disintegration is a not a ‘compositional’ operation that can be obtained by combining other string diagrammatic primitives. The reason is that disintegration involves normalisation, via division in (3.17). Still it would be nice to be able to use disintegration in diagrammatic form. For this purpose one may use a trick: in the setting of Definition 3.5.4 we ‘bend’ the relevant wire downwards and put a gray box around the result in order to suggest that its interior is closed off and has become inaccessible. Thus, we write the disintegration of:

![Bent Box Diagram]

This ‘shaded box’ notation can also be used for more complicated forms of disintegration, as described above.

Exercises

3.5.1 1 Prove that if a joint state has full support, then each of its marginals has full support.

2 Consider the joint state $\omega = \frac{1}{2}|a,b\rangle + \frac{1}{2}|a',b'\rangle \in D(A \times B)$ for $A = \{a,a'\}$ and $B = \{b,b'\}$. Check that both marginals $\omega[1,0] \in D(A)$ and $\omega[0,1] \in D(B)$ have full support, but $\omega$ itself does not. Conclude that the converse of the previous point does not hold.

3.5.2 Consider the box $f$ in Remark 3.5.6. Write done the equation for the disintegration $f[0,1,0,1,0 \mid 1,0,1,0,1]$. Formulate also what uniqueness means.

3.5.3 Show how to obtain the disintegration $f[0,0,0,1,1 \mid 0,1,1,0,0]$ in Remark 3.5.6 from the formulation in Definition 3.5.4 via rearranging and combining wires (via $\times$).
3.6 Disintegration for states

The previous section has described disintegration for channels, but in the literature it is mostly used for states only, that is, for boxes without incoming wires. This section will look at this special case, following \[13\].

In its most basic form disintegration involves extracting a channel from a (binary) joint state, so that the state can be reconstructed as a graph — as in Definition \[1.8.10\] This works as follows.

From $\omega \in \mathcal{D}(X \times Y)$ we can extract a channel $\omega \mid [0,1] \mid [1,0]$ and also a channel $\omega \mid [1,0] \mid [0,1]$ in the opposite direction, provided that $\omega$’s marginals have full support. The direction of the channels is thus in a certain sense arbitrary, and does not reflect any form of causality; for more information, see \[83, 84, 94\] (or \[50\] for a string-diagrammatic account).

**Example 3.6.1.** We shall look at two examples, involving spaces $A = \{a, a^\perp\}$ and $B = \{b, b^\perp\}$.
Chapter 3. Directed Graphical Models

1. Consider the following state \( \omega \in \mathcal{D}(A \times B) \),
\[
\omega = \frac{1}{4} |a, b\rangle + \frac{1}{2} |a^+, b^+\rangle + \frac{1}{4} |a^-, b^-\rangle.
\]

We have as first marginal \( \sigma := \omega[1,0] = \frac{3}{4} |a\rangle + \frac{1}{2} |a^+\rangle \) with full support. The extracted channel \( c := \omega[0,1 | 1,0] : A \rightarrow B \) is given by:
\[
c(a) = \frac{\omega(a,b)}{\sigma(a)} |b\rangle + \frac{\omega(a,b^+)}{\sigma(a)} |b^+\rangle = \frac{1}{4} |b\rangle + \frac{1}{2} |b^+\rangle
c(a^+) = \frac{\omega(a^+,b)}{\sigma(a^+)} |b\rangle + \frac{\omega(a^+,b^+)}{\sigma(a^+)} |b^+\rangle = \frac{1}{4} |b\rangle + \frac{1}{4} |b^+\rangle = \frac{1}{2} |b^+\rangle.
\]

Then indeed, \( \text{gr}(\sigma, c) = (\text{id}, c) \gg \sigma = \omega \).

2. Now let’s start from:
\[
\omega = \frac{1}{3} |a, b\rangle + \frac{2}{3} |a, b^+\rangle.
\]

Then \( \sigma := \omega[1,0] = 1 |a\rangle \). It does not have full support. Let \( \tau = \mathcal{D}(B) \) be an arbitrary state. We define \( c : A \rightarrow B \) as:
\[
c(a) = \frac{1}{2} |b\rangle + \frac{1}{2} |b^+\rangle \quad \text{and} \quad c(a^+) = \tau.
\]

We then still get \( (\text{id}, c) \gg \sigma = \omega \), no matter what \( \tau \) is.

More generally, if we don’t have full support, disintegrations still exist, but they are not unique. We generally avoid such non-uniqueness by requiring full support.

3.6.1 Disintegration of states and conditioning

With disintegration (of states) we can express conditioning (updating) of states in a new way. Also, the earlier results on crossover inference can now be reformulated via conditioning.

For the next result, recall that we can identify a predicate \( X \rightarrow [0, 1] \) with a channel \( X \rightarrow 2 \), see Exercise 2.4.8.

Proposition 3.6.2. Let \( \omega \in \mathcal{D}(A) \) be state and \( p \in \text{Pred}(A) \) be predicate, both on the same finite set \( A \). Assume that the validity \( \omega \models p \) is neither 0 nor 1.

Then we can express the conditionings \( \omega_p, \omega_{p^+} \in \mathcal{D}(A) \) as

168
3.6. Disintegration for states

(interpreted) string diagrams:

\[ \omega|_p = \]

\begin{align*}
&\begin{array}{c}
A \\
\downarrow \\
\downarrow p \\
\downarrow \\
\downarrow \omega \\
\downarrow \\
\downarrow 2 \\
\downarrow 1 \\
\end{array}
&\begin{array}{c}
A \\
\downarrow \\
\downarrow p \\
\downarrow \\
\downarrow \omega \\
\downarrow \\
\downarrow 2 \\
\downarrow 0 \\
\end{array}
\end{align*}

and \[ \omega|_{p^+} = \]

\begin{align*}
&\begin{array}{c}
A \\
\downarrow \\
\downarrow p \\
\downarrow \\
\downarrow \omega \\
\downarrow \\
\downarrow 2 \\
\downarrow 0 \\
\end{array}
&\begin{array}{c}
A \\
\downarrow \\
\downarrow p \\
\downarrow \\
\downarrow \omega \\
\downarrow \\
\downarrow 2 \\
\downarrow 1 \\
\end{array}
\end{align*}

**Proof.** First, let’s write \( \sigma := (p, \text{id}) \gg \omega \in \mathcal{D}(2 \times A) \) for the joint state that is disintegrated in the above string diagrams. The pre-conditioning for disintegration is that the marginal \( \sigma[1, 0] \in \mathcal{D}(2) \) has full support. Explicitly, this means for \( b \in 2 = \{0, 1\}, \)

\[ \sigma[1, 0](b) = \sum_{a \in A} \sigma(b, a) = \sum_{a \in A} \omega(a) \cdot p(a)(b) = \begin{cases} \omega \models p & \text{if } b = 1 \\ \omega \models p^\perp & \text{if } b = 0. \end{cases} \]

The full support requirement that \( \sigma[1, 0](b) > 0 \) for each \( b \in 2 \) means that both \( \omega \models p \) and \( \omega \models p^\perp = 1 - (\omega \models p) \) are non-zero. This holds by assumption.

We elaborate the above string diagram on the left. Let’s write \( c \) for the extracted channel. It is, according to (3.20),

\[ c(1) = \sum_a \frac{\sigma(1, a)}{\sum_a \sigma(1, a)} |a\rangle = \sum_a \frac{\omega(a) \cdot p(a)}{\omega \models p} |a\rangle = \omega|_p. \]

We move to crossover inference, as described in Corollary 2.5.9. There, a joint state \( \omega \in \mathcal{D}(X \times Y) \) is used, which can be described as a graph \( \omega = \text{gr}(\sigma, c) \). But we can drop this graph assumption now, since it can be obtained from \( \omega \) via disintegration — assuming that \( \omega \)’s first marginal has full support.

Thus we come to the following reformulation of Corollary 2.5.9.

**Theorem 3.6.3.** Let \( \omega \in \mathcal{D}(X \times Y) \) be joint state whose first marginal \( \omega[1, 0] \in \mathcal{D}(X) \) has full support, so that the channel \( \omega[0, 1 \parallel 1, 0] : X \rightarrow Y \) exists by disintegration, and thus \( \omega = \text{gr}(\omega[1, 0], \omega[0, 1 \parallel 1, 0]) \). Then:

1. for a factor \( p \) on \( X \),

\[ (\omega|_{p^\perp})[0, 1] = \omega[0, 1 \parallel 1, 0] \gg (\omega[1, 0]|_p). \]

2. for a factor \( q \) on \( Y \),

\[ (\omega|_{q^\perp})[1, 0] = \omega[1, 0]|_{\omega[0, 1 \parallel 1, 0] \ll q}. \]
3.6.1 Let $\sigma \in \mathcal{D}(X)$ have full support and consider $\omega := \sigma \otimes \tau$ for some $\tau \in \mathcal{D}(Y)$. Check that the channel $X \rightarrow Y$ extracted from $\omega$ by disintegration is the constant function $x \mapsto \tau$. Give a string diagrammatic account of this situation.

3.6.2 Consider an extracted channel $\omega[1,0,0|0,0,1]$ for some state $\omega$.

1 Write down the defining equation for this channel, as string diagram.

2 Check that $\omega[1,0,0|0,0,1]$ is the same as $\omega[1,0,1][1,0|0,1]$.

3.6.3 Check that a marginalisation $\omega M$ can also be described as disintegration $\omega[M|0,\ldots,0]$ where the number of 0’s equals the length of the mask/list $M$.

3.6.4 Disintegrate the distribution $Flm(\tau) \in \mathcal{D}([H,L] \times \{1,2,3\})$ in Example 4.1.2 to a channel $(H,L) \rightarrow \{1,2,3\}$.

3.7 Bayesian inversion

Bayesian inversion is a mechanism for turning a channel $A \rightarrow B$ into a channel $B \rightarrow A$ in the opposite direction, in presence of a state on $A$. Bayesian inversion can be defined via disintegration, but also in the other direction, disintegration can be defined via Bayesian inversion. Thus, Bayesian inversion and disintegration are interdefinable.

This section introduces the basics of Bayesian inversion, following [15] and [13]. A categorical analysis of Bayesian inversion is given in the next section. Bayesian inversion will be used later on in Jeffrey’s adaptation rule and in learning.

**Definition 3.7.1.** Let $c: A \rightarrow B$ be a channel with a state $\omega \in \mathcal{D}(A)$ such that $c \gg \omega \in \mathcal{D}(B)$ has full support. The **Bayesian inversion** of $c$ wrt. $\omega$ is a channel written as $c^*_\omega : B \rightarrow A$, and defined as disintegration of the joint state
3.7. Bayesian inversion

\[ \langle c, \text{id} \rangle \gg \omega \ \text{in:} \]

\[ c^\dagger \omega \]

such that

\[ \sum_{a} \omega(a) \cdot c(a)(b) \]

(3.21)

By construction, \( c^\dagger \omega \) is the unique box giving the above equation. By applying \( \mp \) to the left outgoing line on both sides of the above equation we get:

\[ \omega = c^\dagger \omega \gg (c \gg \omega). \] (3.22)

The superscript-dagger notation \((-)^\dagger\) is used since Bayesian inversion is similar to the adjoint-transpose of a linear operator \( A \) between Hilbert spaces, see [15] (and Exercise 3.7.6 below). This transpose is typically written as \( A^\dagger \), or also as \( A^* \). The dagger notation is more common in quantum theory, and has been formalised in terms of dagger categories [16]. This categorical approach is sketched in Section 3.8 below.

We know from Proposition 3.5.5 that disintegrations exist, so Bayesian inversions also exist, as special case. The next result gives a concrete description, in terms of backward inference with a point predicate.

**Proposition 3.7.2.** For a channel \( c: A \to B \) and state \( \omega \in D(A) \) such that \( c \gg \omega \) has full support, one has:

\[ c^\dagger_{\omega}(b) = \omega|_{c \gg \omega}(a) = \sum_{a} \omega(a) \cdot c(a)(b) \cdot (c \gg \omega)(a). \] (3.23)

**Proof.** By uniqueness of Bayesian inversion, it suffices to show that the formulation (3.23) ensures that the equation on the right in (3.21) holds. This is easy:

\[ ((\text{id}, c^\dagger_{\omega}) \gg (c \gg \omega))(b, a) = c^\dagger_{\omega}(b)(a) \cdot (c \gg \omega)(a) \]

\[ = \omega(a) \cdot c(b)(b) \cdot (c \gg \omega)(a) \]

\[ = \omega(a) \cdot c(a)(b) \]

\[ = ((c, \text{id}) \gg \omega)(b, a). \]

Now that we know that Bayesian inversion has this backward-inference-with-point-evidence form, we can recognise that we have been using several times it before. We briefly review some of these examples.
Example 3.7.3. 1 In Example 2.5.2 we used a channel \( c : \{H,T\} \rightarrow \{W,B\} \) from sides of a coin to colors of balls in an urn, together with a fair coin \( \gamma \in D(\{H,T\}) \). We had evidence of a white ball and wanted to know the updated coin distribution. The outcome that we computed can be described via a dagger channel, since:

\[
c^\dagger(W) = \gamma \cdot \chi = (\frac{1}{2}\lvert H \rangle + \frac{1}{2}\lvert T \rangle).
\]

The same redescription in terms of Bayesian inversion can be used for Examples 2.5.3 and 2.5.6. In Example 2.5.4 we can use Bayesian inversion for the point evidence case, but not for the illustration with soft evidence, i.e. with fuzzy predicates. This matter will be investigated further in Section 3.11.

2 In Definition 3.4.4 we have seen filtering for hidden Markov models, starting from a sequence of factors \( \vec{p} \) as observations. In practice these factors are often point predicates \( \chi_y \) for a sequence of elements \( \vec{y} \) of the visible space \( Y \). In that case one can describe filtering via Bayesian inversion as follows. Suppose we have a hidden Markov model with transition channel \( t : X \rightarrow X \), emission channel \( e : X \rightarrow Y \) and initial state \( \sigma \in D(X) \). The filtered sequence of states for point observations \( \vec{y} \) in \( Y \) is given by:

\[
\sigma_1 = \sigma \quad \text{and} \quad \sigma_{i+1} = t \gg (\sigma_i \cdot \chi_{y_i}) = t \gg e^\tau_{y_i}(y_i) = (t \cdot e^\tau_{y_i})(y_i).
\]

Above, we have defined Bayesian inversion in terms of disintegration. In fact, the two notions are inter-definable, since disintegration can also be defined in terms of Bayesian inversion. This will be shown next.

Lemma 3.7.4. Let \( \omega \in D(A \times B) \) be such that its first marginal \( \pi_1 \gg \omega = \omega^{[1,0]} \in D(A) \) has full support, where \( \pi_1 : A \times B \rightarrow A \) is the projection channel. The second marginal of the Bayesian inversion:

\[
\pi_1^{[0,1]} \quad \text{for} \quad (\pi_1)^\tau : A \rightarrow A \times B,
\]

is then the disintegration channel \( A \rightarrow B \) for \( \omega \).

Proof. We first note that the projection channel \( \pi_1 : A \times B \rightarrow A \) can be written as string diagram:

\[
\pi_1 = \begin{array}{c}
\vdots
\end{array}
\]
3.7. Bayesian inversion

We need to prove the equation in (3.18). It can be obtained via purely diagrammatic reasoning:

\[ \omega (\pi_1)^\omega = \omega \]

The first equation is an instance of (3.21).

We illustrate the use of both disintegration and Bayesian inversion in an example of ‘naive’ Bayesian classification from [101]; we follow the analysis of [13].

**Example 3.7.5.** Consider the table in Figure 3.1. It collects data about certain weather conditions and whether or not there is playing (outside). The question asked in [101] is: given this table, what can be said about the probability of playing if the outlook is Sunny, the temperature is Cold, the humidity is High and it is Windy? This is a typical Bayesian update question, starting from (point) evidence. We will first analyse the situation in terms of channels.

We start by extracting the underlying spaces for the columns/categories in the table in Figure 3.1. We choose obvious abbreviations for the entries in the table:

\[ O = \{s, o, r\} \quad T = \{h, m, c\} \quad H = \{h, n\} \quad W = \{t, f\} \quad P = \{y, n\} \]
These sets are joined into a single product space:

\[ S := O \times T \times H \times W \times P. \]

It combines the five columns in Figure 3.1. The table itself can now be considered as a multiset in \( M(S) \) with 14 elements, each with multiplicity one. We will turn it immediately into an empirical distribution — formally via frequentist learning. It yields a distribution \( \tau \in \mathcal{D}(S) \), with 14 entries, each with the same probability, written as:

\[
\tau = \frac{1}{14} \langle s, h, h, f, n \rangle + \frac{1}{14} \langle s, h, t, n \rangle + \cdots + \frac{1}{14} \langle r, m, h, t, n \rangle.
\]

We use a ‘naive’ Bayesian model in this situation, which means that we assume that all weather features are independent. This assumption can be visualised via the following string diagram:

This model oversimplifies the situation, but still it often leads to good (enough) outcomes.

We take the above perspective on the distribution \( \tau \), that is, we ‘factorise’ \( \tau \) according to this string diagram (3.24). Obviously, the play state \( \pi \in \mathcal{D}(P) \) is obtained as the last marginal:

\[
\pi := \tau[0, 0, 0, 0, 1] = \frac{9}{14} |s\rangle + \frac{5}{14} |n\rangle.
\]

Next, we extract four channels \( c_O, c_T, c_H, c_W \) via appropriate disintegrations, from the Play column to the Outlook / Temperature / Humidity / Windy columns.

\[
\begin{align*}
  c_O & := \tau[1, 0, 0, 0, 0 | 0, 0, 0, 0, 1] & c_H & := \tau[0, 0, 1, 0, 0 | 0, 0, 0, 1, 0] \\
  c_T & := \tau[0, 1, 0, 0, 0 | 0, 0, 0, 1, 0] & c_W & := \tau[0, 0, 0, 1, 0 | 0, 0, 0, 0, 1].
\end{align*}
\]

For instance the ‘outlook’ channel \( c_O: P \to O \) looks as follows.

\[
c_O(y) = \frac{3}{7} |s\rangle + \frac{2}{7} |o\rangle + \frac{2}{7} |r\rangle \quad c_O(n) = \frac{2}{7} |s\rangle + \frac{5}{7} |r\rangle.
\]

(3.25)

It is analysed in greater detail in Exercise 3.7.3 below.

Now we can form the tuple channel of these extracted channels, called \( e \) in:

\[
P \xrightarrow{e=(c_O,c_T,c_H,c_W)} O \times T \times H \times W
\]

Recall the question that we started from: what is the probability of playing
if the outlook is Sunny, the temperature is Cold, the humidity is High and it is Windy? These features can be translated into an element \((s, c, h, w)\) of the codomain \(O \times T \times H \times W\) of this tuple channel — and thus into a point predicate. Hence our answer can be obtained by Bayesian inversion of the tuple channel, as:

\[
c^\dagger(s, c, h, t) \bigg|_{c\ll 1, h\ll 0} = \frac{125}{611} y + \frac{486}{611} n
\]

This corresponds to the probability 20.5% calculated in [101] — without any disintegration or Bayesian inversion.

In the end one can reconstruct a joint state with space \(S\) via the extracted channel, as graph:

\[
\langle c_O, c_T, c_H, c_W, \text{id} \rangle \gg \pi.
\]

This state differs considerably from the original table/state \(\tau\). It shows that the shape (3.24) does not really fit the data that we have in Figure 3.1. But recall that this approach is called naive. We shall soon look closer into such matters of shape in Section 3.9.

After this concrete application of Bayesian inversion we turn to some of its more abstract properties.

**Lemma 3.7.6.** For channels \(c: X \rightarrow Y\) and \(d: X \rightarrow Z\) we have:

\[
\langle c, d \rangle^\dagger(y, z) = \omega|_{(c \ll 1) \& (d \ll 1)} = d^\dagger_{c_{(y)}} (z)
\]

\[
= \omega|_{(d \ll 1) \& (c \ll 1)} = c^\dagger_{d_{(z)}} (y).
\]

**Proof.** We use the explicit formulation (3.23) of Bayesian inversion in:

\[
\langle c, d \rangle^\dagger(y, z) = \omega|_{(c \ll 1) \& (d \ll 1)} = \omega|_{\Delta = 1(c \ll 1) \& (d \ll 1)} = \omega|_{\Delta = 1(c \ll 1) \& (d \ll 1)} = \omega|_{c \ll 1 \& d \ll 1} = c^\dagger_{d_{(z)}} (y) = d^\dagger_{c_{(y)}} (z).
\]

The outcome \(c^\dagger_{d_{(z)}} (y)\) can be obtained by using that conjunction \& is commutative in this proof. 

3.7.1 Bayesian inversion and inference

In Definition 2.5.1 we have carefully distinguished forward inference \(c \gg (\omega|_p)\) from backward inference \(\omega|_{c \ll q}\). Since Bayesian inversion turns channels around, the question arises whether it also turns inference around: can
Chapter 3. Directed Graphical Models

one express forward (resp. backward) inference along $c$ in terms of backward (resp. forward) inference along the Bayesian inversion of $c$. The next result from [49] shows that this is indeed the case. It demonstrates that the directions of inference and of channels are intimately connected.

**Theorem 3.7.7.** Let $c: X \to Y$ be a channel with a state $\omega \in \mathcal{D}(X)$ on its domain, such that the transformed state $\sigma := c \gg \omega$ has full support.

1. Given a factor $q$ on $Y$, we can express backward inference as forward inference via:

$$\omega_{\mid c} \gg (\sigma_{\mid q}).$$

2. Given a factor $p$ on $X$, we can express forward inference as backward inference:

$$c \gg (\omega_{\mid p}) = \sigma_{\mid c \ll p}.$$

**Proof.** 1 For $x \in X$ we have:

$$\left( c_{\mid c} \gg (\sigma_{\mid q}) \right)(x) = \sum_y c_{\mid c}(y)(x) \cdot (c \gg \omega)_{\mid q}(y)$$

$$= \sum_y \frac{c(x)(y) \cdot \omega(x) \cdot (c \gg \omega)(y) \cdot q(y)}{\omega(x) \cdot (\sum_y c(x)(y) \cdot q(y)) \cdot (c \gg \omega)_{\mid q}}$$

$$= \frac{\omega(x) \cdot (c \ll q)(x)}{\omega \gg c \ll q}$$

$$= \omega_{\mid c \ll q}(x).$$

2. Similarly, for $y \in Y$,

$$\left( \sigma_{\mid c \ll p} \right)(y) = \frac{(c \gg \omega)(y) \cdot (c_{\mid c} \ll p)(y)}{c \gg \omega \gg c_{\mid c} \ll p}$$

$$= \sum_x \frac{(c \gg \omega)(y) \cdot c_{\mid c}(y)(x) \cdot p(x)}{(c \gg \omega)_{\mid c} \gg \omega \gg (c \gg \omega)(y) \cdot p(x)}$$

$$= \sum_x \frac{\omega(x) \cdot \omega_{\mid p}(x)}{\omega \gg p}$$

$$= \sum_x \omega(x) \cdot \omega_{\mid p}(x)$$

$$= \left( c \gg (\omega_{\mid p}) \right)(y).$$

We conclude with a follow-up of the crossover update corollary [2.5.9] There
3.7. Bayesian inversion

we look at marginalisation after update. Here we look at extraction of a channel, namely at the same position of the original channel. It can be expressed in terms of an updated channel, see Definition 2.3.1(3).

**Theorem 3.7.8.** Let \( c : X \rightarrow Y \) be a channel with state \( \sigma \in \mathcal{D}(X) \) on its domain, and let \( p \in \text{Fact}(X) \) and \( q \in \text{Fact}(Y) \) be factors.

1. The extraction on an updated graph state yields:
   \[
   (\langle \text{id}, c \rangle \gg \sigma)|_{p \otimes q}[0, 1 \mid 1, 0] = c|_q \quad \text{where} \quad c|_q(x) := c(x)|_q.
   \]

2. And as a result:
   \[
   (\langle \text{id}, c \rangle \gg \sigma)|_{p \otimes q} = (\langle \text{id}, c \rangle \gg \sigma)|_{p \otimes (c \otimes q)}.
   \]

**Proof.** 1. For \( x \in X \) and \( y \in Y \) we have:

\[
(\langle \text{id}, c \rangle \gg \sigma)|_{p \otimes q}[0, 1 \mid 1, 0](x)(y)
\]

\[
= (\langle \text{id}, c \rangle \gg \sigma)|_{p \otimes q}(x, y)
\]

\[
= \sum_x (\langle \text{id}, c \rangle \gg \sigma)(x, v) \cdot (p \otimes q)(x, v)
\]

\[
= \sum_x \sigma(x) \cdot c(x)(y) \cdot p(x) \cdot q(y)
\]

\[
= \sum_x c(x)(y) \cdot q(y)
\]

\[
= c(x)|_q(y)
\]

2. A joint state like \( (\langle \text{id}, c \rangle \gg \sigma)|_{p \otimes q} \) can always be written as graph \( \text{gr}(\tau, d) = \langle \text{id}, d \rangle \gg \tau \). The channel \( d \) is obtained by distintegration of the joint state, and equals \( c|_q \) by the previous point. The state \( \tau \) is the first marginal of the joint state. Hence we are done by:

\[
(\langle \text{id}, c \rangle \gg \sigma)|_{p \otimes q}[1, 0]
\]

\[
= (\langle \text{id}, c \rangle \gg \sigma)|_{[1 \otimes q] \otimes [1 \otimes p]}[1, 0] \quad \text{by Exercise 2.4.5}
\]

\[
= (\langle \text{id}, c \rangle \gg \sigma)|_{1 \otimes q}[1, 0] \quad \text{by Lemma 2.3.5(3)}
\]

\[
= (\langle \text{id}, c \rangle \gg \sigma)|_{1 \otimes q}[1, 0] \quad \text{by Lemma 2.3.5(6)}
\]

\[
= \sigma|_{c \otimes q} \quad \text{by Corollary 2.5.9(2)}
\]

\[
= \sigma|_{p \otimes (c \otimes q)} \quad \text{by Lemma 2.3.5(3)}. \quad \square
\]

177
In Section 3.11 we will use Bayesian inversion for a new form of belief revision, called Jeffrey’s adaptation.

**Exercises**

3.7.1 Check that Proposition 3.6.2 can be reformulated as:

\[ \omega|_p = p^\omega (1) \quad \text{and} \quad \omega|_{p^\perp} = p^\omega (0). \]

3.7.2 1 Prove Equation (3.22) concretely, using the backward inference formulation (3.23) of Bayesian inversion.

2 Check that Equation (3.22) can be obtained as special case of Theorem 3.7.7 (1), namely when \( q \) is the truth predicate 1.

3.7.3 Write \( \sigma \in M(O \times T \times H \times W \times P) \) for the weather-play table in Figure 3.1, as multidimensional multiset.

1 Compute the marginal multiset \( \sigma\left[1, 0, 0, 0, 1\right] \in M(O \times P) \).

2 Reorganise this marginalised multiset as a 2-dimensional table with only Outlook (horizontal) and Play (vertical) data, as given below, and check how this ‘marginalised’ table relates to the original one in Figure 3.1.

<table>
<thead>
<tr>
<th></th>
<th>Sunny</th>
<th>Overcast</th>
<th>Rainy</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>no</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

3 Deduce a channel \( P \rightarrow O \) from this table, and compare it to the description (3.25) of the channel \( c_O \) given in Example 3.7.5 — see also Lemma 1.6.2 and Proposition 4.1.1.

4 Do the same for the marginal tables \( \sigma\left[0, 1, 0, 0, 1\right], \sigma\left[0, 1, 0, 1, 1\right], \sigma\left[0, 0, 0, 1, 1\right]\), and the corresponding channels \( c_T, c_H, c_W \) in Example 3.7.5.

3.7.4 Prove that the equation

\[ ((id, c) \gg \sigma)|_{p\otimes} [0, 1] \mid 1, 0] = c. \]

can be obtained both from Theorem 3.7.7 and from Theorem 3.7.8.

3.7.5 Prove that for a state \( \omega \in D(X \times Y) \) with full support one has:

\[ \omega[1, 0] \mid 0, 1] = \left(\omega[0, 1] \mid 1, 0]\right)^\top : Y \rightarrow X. \]

3.7.6 For \( \omega \in D(X), c : X \rightarrow Y, p \in \text{Obs}(X) \) and \( q \in \text{Obs}(Y) \) prove:

\[ c \gg \omega \Vdash (c_\omega \ll p) \& q = \omega \Vdash p \& (c \ll q). \]
This is a reformulation of [15, Thm. 6]. If we ignore the validity signs \( \models \) and the states before them, then we can recognise in this equation the familiar ‘adjointness property’ of daggers (adjoint transposes) in the theory of Hilbert spaces: \( (A^\dagger(x) \mid y) = (x \mid A(y)) \).

3.7.7 Recall from Exercise 1.7.9 what it means for \( c : X \rightarrow Y \) to be a bi-channel. Check that \( c^* : Y \rightarrow Y \), as defined there, is \( c^\dagger \nu \), where \( \nu \) is the uniform distribution on \( X \) (assuming that \( X \) is finite).

### 3.8 Categorical aspects of Bayesian inversion

As suggested in the beginning of this section the ‘dagger’ of a channel — *i.e.* its Bayesian inversion — can also be described categorically. It turns out to be a special ‘dagger’ functor. Such reversal is quite common for non-deterministic computation, see Example 3.8.1 below. The fact that this same abstract structure exists for probabilistic computation demonstrates once again that Bayesian inversion is a canonical operation — and that category theory provides a useful language for making such similarities explicit.

This section goes a bit deeper into the category-theoretic aspects of probabilistic computation in general and of Bayesian inversion in particular. It is not essential for the rest of this book, but provides deeper insight into the underlying structures.

**Example 3.8.1.** Recall the category \( \text{Chan}(\mathcal{P}) \) of non-deterministic computations. Its objects are sets \( X \) and its morphisms \( f : X \rightarrow Y \) are functions \( f : X \rightarrow \mathcal{P}(X) \). The identity morphism \( \text{unit} : X \rightarrow X \) in \( \text{Chan}(\mathcal{P}) \) is the singleton function \( \text{unit}(x) = \{x\} \). Composition of \( f : X \rightarrow Y \) and \( g : Y \rightarrow Z \) is the function \( g \circ f : X \rightarrow Z \) given by:

\[
(g \circ f)(x) = \{z \in Z \mid \exists y \in Y, y \in f(x) \text{ and } z \in g(y)\}.
\]

It is not hard to see that \( \circ \) is associative and has \( \text{unit} \) as neutral element. In fact, this has already been proven more generally, in Lemma 1.7.3.

There are two aspects of the category \( \text{Chan}(\mathcal{P}) \) that we wish to illustrate, namely (1) that it has an ‘inversion’ operation, in the form of a dagger functor, and (2) that it is isomorphic to the category \( \text{Rel} \) of sets with relations between them (as morphisms). Probabilistic analogues of these two points will be described later.

1 We start from a very basic observation, namely that morphisms in \( \text{Chan}(\mathcal{P}) \) can be reversed. There is a bijective correspondence, indicated by the double
We write $\text{Rel}$ in $\text{F}$ and $\text{Chan}$.

This correspondence sends $f: X \to \mathcal{P}(Y)$ to the function $f^{\dagger}: Y \to \mathcal{P}(X)$ with $f^{\dagger}(y) = \{x \mid y \in f(x)\}$. Hence $y \in f(x)$ iff $x \in f^{\dagger}(y)$. Similarly one sends $g: Y \to \mathcal{P}(X)$ to $g^{\dagger}: X \to \mathcal{P}(Y)$ via $g^{\dagger}(x) = \{y \mid x \in g(y)\}$. Clearly, $f^{\dagger \dagger} = f$ and $g^{\dagger \dagger} = g$.

It turns out that this dagger operation $(-)^{\dagger}$ interacts nicely with composition: one has $\text{unit}^{\dagger} = \text{unit}$ and also $(g \circ f)^{\dagger} = f^{\dagger} \circ g^{\dagger}$. This means that the dagger is functorial. It can be described as a functor $(-)^{\dagger}: \text{Chan}(\mathcal{P}) \to \text{Chan}(\mathcal{P})^{op}$, which is the identity on objects: $X^{\dagger} = X$. The opposite $(\cdot)^{op}$ category is needed for this functor since it reverses arrows.

2 We write $\text{Rel}$ for the category with sets $X$ as objects and relations $R \subseteq X \times Y$ as morphisms $X \to Y$. The identity $X \to X$ is given by the equality relation $\mathcal{E}q_X \subseteq X \times X$, with $\mathcal{E}q_X = \{(x, x) \mid x \in X\}$. Composition of $R \subseteq X \times Y$ and $S \subseteq X \times Z$ is the ‘relational’ composition $S \bullet R \subseteq X \times Z$ given by:

$$S \bullet R := \{(x, z) \mid \exists y \in Y. R(x, y) \text{ and } S(y, z)\}.$$ 

It is not hard to see that we get a category in this way.

There is a ‘graph’ functor $G: \text{Chan}(\mathcal{P}) \to \text{Rel}$, which is the identity on objects: $G(X) = X$. On a morphism $f: X \to Y$, that is, on a function $f: X \to \mathcal{P}(Y)$, we define $G(f) \subseteq X \times Y$ to be $G(f) = \{(x, f(x)) \mid x \in X\}$. Then: $G(\text{unit}_X) = \mathcal{E}q_X$ and $G(\text{ID}_X f) = G(f) \circ G(f)$.

In the other direction there is also a functor $F: \text{Rel} \to \text{Chan}(\mathcal{P})$, which is again the identity on objects: $F(X) = X$. On a morphism $R: X \to Y$ in $\text{Rel}$, that is, on a relation $R \subseteq X \times Y$, we define $F(R): X \to \mathcal{P}(Y)$ as $F(R)(x) = \{y \mid R(x, y)\}$. This $F$ preserves identities and composition.

These two functors $G$ and $F$ are each other’s inverses, in the sense that:

$$F \circ G = \text{id}: \text{Chan}(\mathcal{P}) \to \text{Chan}(\mathcal{P}) \quad \text{and} \quad G \circ F = \text{id}: \text{Rel} \to \text{Rel}.$$ 

This establishes an isomorphism $\text{Chan}(\mathcal{P}) \cong \text{Rel}$ of categories.

Interestingly, $\text{Rel}$ is also a dagger category, via the familiar operation of reversal of relations: for $R \subseteq X \times Y$ one can form $R^{\dagger} \subseteq Y \times X$ via $R^{\dagger}(y, x) = R(x, y)$. This yields a functor $(-)^{\dagger}: \text{Rel} \to \text{Rel}^{op}$, obviously with $(-)^{\dagger \dagger} = \text{id}$.

Moreover, the above functors $G$ and $F$ commute with the daggers of $\text{Chan}(\mathcal{P})$ and $\text{Rel}$, in the sense that:

$$G(f^{\dagger}) = G(f)^{\dagger} \quad \text{and} \quad F(R^{\dagger}) = F(R)^{\dagger}. $$

180
3.8. Categorical aspects of Bayesian inversion

We shall prove the first equation and leave the second one to the interested reader. The proof is obtained by carefully unpacking the right definition at each stage. For a function \( f : X \to \mathcal{P}(Y) \) and elements \( x \in X, y \in Y \),

\[
G(f^\dagger)(y, x) \iff x \in f^\dagger(y) \iff y \in f(x) \iff G(f)(x, y) \iff G(f)^\dagger(y, x).
\]

We now move from non-deterministic to probabilistic computation. Our aim is to obtain analogous results, namely inversion in the form of a dagger functor on a category of probabilistic channels, and an isomorphism of this category with a category of probabilistic relations. One may expect that these results hold for the category \( \text{Chan}(D) \) of probabilistic channels. But the situation is a bit more subtle. Recall from the previous section that the dagger (Bayesian inversion) \( c^\dagger_\omega : Y \to X \) of a probabilistic channel \( c : X \to Y \) requires a state \( \omega \in D(X) \) on the domain. In order to conveniently deal with this situation we incorporate these states \( \omega \) into the objects of our category. We follow [15] and denote this category as \( \text{Krn} \); its morphisms are ‘kernels’.

Definition 3.8.2. The category \( \text{Krn} \) of kernels has:

- objects: pairs \((X, \sigma)\) where \( X \) is a finite set and \( \sigma \in D(X) \) is a distribution on \( X \) with full support;
- morphisms: \( f : (X, \sigma) \to (Y, \tau) \) are probabilistic channels \( f : X \to Y \) with \( f \gg \sigma = \tau \).

Identity maps \((X, \sigma) \to (X, \sigma)\) in \( \text{Krn} \) are identity channels \( \text{unit} : X \to X \), given by \( \text{unit}(x) = 1\_\{x\} \), which we write simply as \( \text{id} \). Composition in \( \text{Krn} \) is ordinary composition \( \circ \) of channels.

Theorem 3.8.3. Bayesian inversion forms a dagger functor \((-)^\dagger : \text{Krn} \to \text{Krn}^\circ \) which is the identity on objects and which sends:

\[
\left( (X, \sigma) \xrightarrow{f} (Y, \tau) \right) \mapsto \left( (Y, \tau) \xleftarrow{f^\dagger} (X, \sigma) \right)
\]

This functor is its own inverse: \( f^{\dagger\dagger} = f \).

Proof. We first have to check that the dagger functor is well-defined, i.e. that the above mapping yields another morphism in \( \text{Krn} \). This follows from (3.22):

\[
f^\dagger \gg \tau = f^\dagger \gg (f \gg \sigma) = \sigma.
\]

Aside: this does not mean that \( f^\dagger \circ f = \text{id} \).

We next show that the dagger preserves identities and composition. We use the concrete formulation of Bayesian inversion of Proposition 3.7.2. A string
diagrammatic proof is also an option, see below. The identity map \( id : (X, \sigma) \rightarrow (X, \sigma) \) in \( \text{Krn} \) satisfies:

\[
\text{id}^\dagger_{\sigma}(x) = \sigma|_{\text{id}=1} = \sigma|_{1} = 1|x \quad \text{see Exercise 2.3.4}
\]

\[= \text{id}(x).\]

For morphisms \((X, \sigma) \xrightarrow{f} (Y, \tau) \xrightarrow{g} (Z, \rho)\) in \( \text{Krn} \) we have, for \( z \in Z \),

\[
(g \circ f)^\dagger_{\sigma}(z) = \sigma|_{(g \circ f)\leq 1} = \sigma|_{f \leq 1} \gg (f \gg \tau)|_{g \leq 1} \quad \text{by Theorem 3.7.7 (1)}
\]

\[= f \gg (\sigma|_{1}) \gg g_{1}(z)
\]

\[= (f_{1} \circ g_{1})(z).
\]

Finally we have:

\[
(f_{\sigma})_{1}(x) = \tau|_{f_{\sigma} \leq 1} = (f \gg \sigma)|_{f_{\sigma} \leq 1} \gg 1|x \quad \text{by Theorem 3.7.7 (2)}
\]

\[= f \gg 1|x \gg \tau = f(x). \quad \square
\]

One can also prove this result via equational reasoning with string diagrams.

For instance preservation of composition \( \circ \) by the dagger follows by uniqueness from:

\[
\begin{array}{c}
\vcenter{\hbox{egin{tikzcd}
& g \\
g \arrow{r} & f \\
\sigma \arrow{u} & \sigma \arrow{u}
\end{tikzcd}}}
= \vcenter{\hbox{egin{tikzcd}
& f_{\sigma} \\
g \arrow{r} & f \\
\sigma \arrow{u} & \sigma \arrow{u}
\end{tikzcd}}}
= \vcenter{\hbox{egin{tikzcd}
& f_{\sigma} \\
g \arrow{r} & f \circ f_{\sigma} \\
\sigma \arrow{u} & \sigma \arrow{u}
\end{tikzcd}}}
= \vcenter{\hbox{egin{tikzcd}
& f_{\sigma} \\
g \arrow{r} & f \circ \tau \\
\sigma \arrow{u} & \sigma \arrow{u}
\end{tikzcd}}}
= \vcenter{\hbox{egin{tikzcd}
& f_{\sigma} \\
g \arrow{r} & f \\
\sigma \arrow{u} & \sigma \arrow{u}
\end{tikzcd}}}
\end{array}
\]

We now turn to probabilistic relations, with the goal of finding a category of such relations that is isomorphic to \( \text{Krn} \). For this purpose we use what are called couplings. The definition that we use below differs only in inessential ways from the formulation that we used in Subsection 2.8.1 for the Wasserstein distance.

**Definition 3.8.4.** We introduce a category \( \text{Cpl} \) of couplings with the same objects as \( \text{Krn} \). A morphism \((X, \sigma) \rightarrow (Y, \tau)\) in \( \text{Cpl} \) is a joint state \( \varphi \in \mathcal{D}(X \times Y) \)
3.8. Categorical aspects of Bayesian inversion

with \( \varphi[1,0] = \sigma \) and \( \varphi[0,1] = \tau \). Such a distribution which marginalises to \( \sigma \) and \( \tau \) is called a coupling between \( \sigma \) and \( \tau \).

Composition of \( \varphi: (X, \sigma) \to (Y, \tau) \) and \( \psi: (Y, \tau) \to (Z, \rho) \) is the distribution \( \psi \bullet \varphi \in D(X \times Z) \) defined as:

\[
(\psi \bullet \varphi) := (\delta[0,1])_{|0,1} \cdot \delta[1,0]_{|1,0} \Rightarrow \tau
\]

\[
= \sum_{x,z} \left( \sum_y \varphi(x,y) \cdot \psi(y,z) \right) \cdot \tau(y) \cdot \delta(x,z).
\]

(3.26)

The identity coupling \( \delta(X,\sigma) : (X, \sigma) \to (X, \sigma) \) is the distribution \( \delta \gg \sigma \).

*The essence of the following result is due to [15], but there it occurs in slightly different form, namely in a setting of continuous probability. Here it is translated to the discrete situation.*

**Theorem 3.8.5.** Couplings as defined above indeed form a category \( \text{Cpl} \).

1 This category carries a dagger functor \((-)^\dagger: \text{Cpl} \to \text{Cpl}^{\text{op}} \) which is the identity on objects; on morphisms it is defined via swapping:

\[
\left((X, \sigma) \to (Y, \tau)\right)^\dagger := \left((Y, \tau) \to (X, \sigma)\right).
\]

More concretely, this dagger is defined by swapping arguments, as in: \( \varphi^\dagger = \sum_{x,y} \varphi(x,y) \cdot \delta(y,x) \).

2 There is an isomorphism of categories \( \text{Krn} \cong \text{Cpl} \), in one direction by taking the graph of a channel, and in the other direction by disintegration. This isomorphism commutes with the daggers on the two categories.

**Proof.** We first need to prove that \( \psi \bullet \varphi \) is a distribution:

\[
\sum_{x,z} (\psi \bullet \varphi)(x,z) = \sum_{x,z} \varphi(x,y) \cdot \psi(y,z) \cdot \tau(y)
\]

\[
= \sum_{y,z} \left( \sum_x \varphi(x,y) \cdot \psi(y,z) \right) \cdot \tau(y)
\]

\[
= \sum_{y,z} \varphi[0,1](y) \cdot \psi(y,z) \cdot \tau(y)
\]

\[
= \sum_{y,z} \tau(y) \cdot \psi(y,z) \cdot \tau(y) = \sum_{y,z} \psi(y,z) = 1.
\]

We leave it to the reader to check that \( \delta(X,\sigma) = \Delta \gg \sigma \) is neutral element for \( \bullet \). We do verify that \( \bullet \) is associative — and thus that \( \text{Cpl} \) is indeed a category.

Let \( \varphi: (X, \sigma) \to (Y, \tau) \), \( \psi: (Y, \tau) \to (Z, \rho) \), \( \chi: (Z, \rho) \to (W, \kappa) \) be morphisms in
\[ (\chi \bullet (\psi \bullet \varphi))(x, w) = \sum_{y,z} \frac{(\psi \bullet \varphi)(x, z) \cdot \chi(z, w) \cdot \rho(z)}{\tau(y) \cdot \rho(z)} = \sum_{y,z} \frac{\varphi(x, y) \cdot \psi(y, z) \cdot \chi(z, w)}{\tau(y)} = \frac{\varphi(x, y) \cdot (\chi \bullet \psi)(y, w)}{\tau(y)} = \left((\chi \bullet \psi) \cdot \varphi\right)(x, w). \]

We turn to the dagger. It is obvious that \((-)^\dagger\) is the identity functor, and also that \((-)^\dagger\) preserves identity maps. It also preserves composition in \(\mathsf{Cpl}\) since:

\[ (\psi \bullet \varphi)^\dagger(z, x) = (\psi \bullet \varphi)(x, z) \]
\[ = \sum_{y} \frac{\varphi(x, y) \cdot \psi(y, z)}{\tau(y)} \]
\[ = \sum_{y} \frac{\varphi^\dagger(z, y) \cdot \psi^\dagger(y, x)}{\tau(y)} = ((\psi^\dagger \bullet \varphi^\dagger)(z, x). \]

The graph operation \(\mathsf{gr}\) on channels from Definition 1.8.10 gives rise to an identity-on-objects ‘graph’ functor \(G: \mathsf{Krn} \to \mathsf{Cpl}\) via:

\[ G(X, \sigma) \xrightarrow{\mathsf{gr}(\sigma, f)} (Y, \tau) \xrightarrow{g} (X, \sigma). \]

where \(\mathsf{gr}(\sigma, f) = \langle \mathsf{id}, f \rangle \Rightarrow \sigma\). This yields a functor since:

\[ G(\mathsf{id}_{(X, \sigma)}) = \langle \mathsf{id}, \mathsf{id} \rangle \Rightarrow \sigma = \mathsf{Eq}_{(X, \sigma)} \]
\[ G(g \circ f)(x, z) = \mathsf{gr}(\mathsf{gr}(\sigma, g \circ f))(x, z) \]
\[ = \sigma(x) \cdot (g \circ f)(x)(z) \]
\[ = \sum_{x} \sigma(x) \cdot f(x)(y) \cdot g(y)(z) \]
\[ = \sum_{x} \sigma(x) \cdot f(x)(y) \cdot \tau(y) \cdot g(y)(z) \]
\[ = \sum_{x} \frac{\mathsf{gr}(\sigma, f)(x, y) \cdot \mathsf{gr}(\tau, g)(y, z)}{\tau(y)} \]
\[ = (\mathsf{gr}(\tau, g) \bullet \mathsf{gr}(\sigma, f))(x, z) \]
\[ = (G(g) \bullet G(f))(x, z). \]

In the other direction we define a functor \(F: \mathsf{Cpl} \to \mathsf{Krn}\) which is the identity on objects and uses disintegration on morphisms: for \(\varphi: (X, \sigma) \to (Y, \tau)\) in \(\mathsf{Cpl}\) one gets a channel \(F(\varphi) := \mathsf{dis}_1(\varphi) = \varphi[0, 1 \mid 1, 0]: X \to Y\) which satisfies, by construction (3.19).

\[ \varphi = \mathsf{gr}(\varphi[1, 0], \mathsf{dis}_1(\varphi)) = \mathsf{gr}(\sigma, F(\varphi)) = GF(\varphi). \]

184
Moreover, $F(\varphi)$ is a morphism $(X, \sigma) \to (Y, \tau)$ in $\textbf{Krn}$ since:

$$F(\varphi) \gg \sigma = \text{dis}_1(\varphi) \gg (\varphi[1, 0])$$
$$= \left(\text{gr}(\varphi[1, 0], \text{dis}_1(\varphi))\right)[0, 1] \stackrel{(3.18)}{=} \varphi[0, 1] = \tau.$$  

We still need to prove that $F$ preserves identities and composition. This follows by uniqueness of disintegration:

\[
\begin{align*}
\langle \text{id}, F(\text{Eq}_{(X, \sigma)}) \rangle & \gg \sigma \\
& = \text{Eq}_{(X, \sigma)} \quad \text{by definition} \\
& = \Delta \gg \sigma \\
& = \langle \text{id}, \text{id} \rangle \gg \sigma \\
\langle \text{id}, F(\psi \bullet \varphi) \rangle & \gg \sigma \\
& = \psi \bullet \varphi \quad \text{by definition} \\
& \stackrel{(3.20)}{=} \langle \varphi[1, 0 \mid 0, 1], \psi[0, 1 \mid 1, 0] \rangle \gg \tau \\
& = (\text{id} \otimes \psi[0, 1 \mid 1, 0]) \gg \left(\langle \varphi[1, 0 \mid 0, 1], \text{id} \rangle \gg (\varphi[1, 0 \mid 0, 1] \gg \sigma)\right) \\
& = (\text{id} \otimes F(\psi)) \gg (\langle F(\varphi)^* , \text{id} \rangle \gg (F(\varphi) \gg \sigma)) \quad \text{by Exercise 3.7.5} \\
& \stackrel{(3.21)}{=} (\text{id} \otimes F(\psi)) \gg (\langle \text{id}, F(\varphi) \rangle \gg \sigma) \\
& = \langle \text{id}, F(\psi) \circ F(\varphi) \rangle \gg \sigma.
\end{align*}
\]

We have seen that, by construction $G \circ F$ is the identity functor on the category $\textbf{Cpl}$. In the other direction we also have $F \circ G = \text{id} : \textbf{Krn} \to \textbf{Krn}$. This follows directly from uniqueness of disintegration.

In the end we like to show that the functors $G$ and $F$ commute with the daggers, on kernels and couplings. This works as follows. First, for $f : (X, \sigma) \to (Y, \tau)$ we have:

$$G(f^\dagger)(y, x) = \text{gr}(\tau, f^\dagger)(y, x) = \tau(y) \cdot f^\dagger(x)(y)$$
$$= (f \gg \sigma)(y) \cdot \frac{\sigma(x) \cdot f(x)(y)}{(f \gg \sigma)(y)}$$
$$= \sigma(x) \cdot f(x)(y)$$
$$= \text{gr}(\sigma, f)(x, y)$$
$$= G(f)(x, y) = G(f)^\dagger(y, x).$$
Chapter 3. Directed Graphical Models

Next, for $\varphi : (X, \sigma) \to (Y, \tau)$ in $\text{Cpl}$,

$$
F(\varphi^\dagger)(y)(x) = \text{dis}_1(\varphi^\dagger)(y)(x) \tag{3.20}
$$

$$
= \varphi^\dagger[1, 0](y) \varphi(x, y) \varphi[0, 1](y) \tag{3.20}
$$

$$
= \varphi[1, 0 \mid 0, 1](y)(x) \tag{3.20}
$$

$$
= (\varphi[0, 1 \mid 1, 0]^\dagger)(y)(x) \quad \text{by Exercise 3.7.5} \tag{3.20}
$$

$$
= F(\varphi)^\dagger(y)(x). \tag{3.20}
$$

We have done a lot of work in order to be able to say that $\text{Krn}$ and $\text{Cpl}$ are isomorphic dagger categories, or, more informally, that there is a one-one correspondence between probabilistic computations (channels) and probabilistic relations.

**Exercises**

3.8.1  Prove in the context of the powerset channels of Example 3.8.1 that:

1. $\text{unit}^\dagger = \text{unit}$ and $(g \circ f)^\dagger = f^\dagger \circ g^\dagger$.
2. $G(\text{unit}_X) = \text{Eq}_X$ and $G(g \circ f) = G(f) \bullet G(f)$.
3. $F(\text{Eq}_X) = \text{unit}_X$ and $F(S \bullet R) = F(S) \circ F(R)$.
4. $(S \bullet R)^\dagger = R^\dagger \bullet S^\dagger$.
5. $F(R^\dagger) = F(R)^\dagger$.

3.8.2  Give a string diagrammatic proof of the property $f^{\dagger\dagger} = f$ in Theorem 3.8.3. Prove also that $(f \otimes g)^\dagger = f^\dagger \otimes g^\dagger$.

3.8.3  Prove the equation $= = $ in the definition (3.26) of composition $\bullet$ in the category $\text{Cpl}$. Give also a string diagrammatic description of $\bullet$.

3.9  Factorisation of joint states

Earlier, in Subsection 1.8.1, we have called a binary joint state non-entwined if it is the product of its marginals. This can be seen as an intrinsic property of the state, which we will express in terms of a string diagram, called its shape. For instance, the state:

$$
\omega = \frac{1}{2}|a, b\rangle + \frac{1}{2}|a, b^\perp\rangle + \frac{1}{12}|a^\perp, b\rangle + \frac{1}{6}|a^\perp, b^\perp\rangle
$$
3.9. Factorisation of joint states

is non-entwined: it is the product of its marginals $\omega[1, 0] = \frac{3}{4}|a\rangle + \frac{1}{4}|a^\perp\rangle$ and $\omega[0, 1] = \frac{1}{3}|b\rangle + \frac{2}{3}|b^\perp\rangle$. We will formulate this as:

\[ \omega \text{ has shape } \begin{array}{c}
\hline \\
\hline \\
\end{array} \begin{array}{c}
\hline \\
\hline \\
\end{array} \\
\text{or as: } \begin{array}{c}
\hline \\
\hline \\
\end{array} \begin{array}{c}
\hline \\
\hline \\
\end{array} \\
\text{and write this as: } \omega \models \begin{array}{c}
\hline \\
\hline \\
\end{array} \begin{array}{c}
\hline \\
\hline \\
\end{array} .\]

We shall give a formal definition of $\models$ below, but at this stage it suffices to read $\sigma \models S$, for a state $\sigma$ and a string diagram $S$, as: there is an interpretation of the boxes in $S$ such that $\sigma = \llbracket S \rrbracket$.

In the above case of $\omega \models \begin{array}{c}
\hline \\
\hline \\
\end{array} \begin{array}{c}
\hline \\
\hline \\
\end{array}$ we obtain $\omega = \omega_1 \otimes \omega_2$, for some state $\omega_1$ that interpretes the box on the left, and some $\omega_2$ interpreting the box on the right. But then:

$\omega[1, 0] = (\omega_1 \otimes \omega_2)[1, 0] = \omega_1$.

Similarly, $\omega[0, 1] = \omega_2$. Thus, in this case the interpretations of the boxes are uniquely determined, namely as first and second marginal of $\omega$.

We conclude that non-entwinedness of an arbitrary binary joint state $\omega$ can be expressed as: $\omega \models \begin{array}{c}
\hline \\
\hline \\
\end{array} \begin{array}{c}
\hline \\
\hline \\
\end{array} $. Here we are interested in similar intrinsic ‘shape’ properties of states and channels that can be expressed via string diagrams. These matters are often discussed in the literature in terms of (conditional) independencies. Here we postpone such independencies and only use string diagrams, for the time being.

In general there may be several interpretations of a string diagram (as shape). Consider for instance:

$\begin{array}{c}
\hline \\
\hline \\
\end{array} \begin{array}{c}
\hline \\
\hline \\
\end{array} \begin{array}{c}
\hline \\
\hline \\
\end{array} \begin{array}{c}
\hline \\
\hline \\
\end{array} \models \begin{array}{c}
\hline \\
\hline \\
\end{array} \begin{array}{c}
\hline \\
\hline \\
\end{array} \\
\text{This state has this form in multiple ways, for instance as: } c_1 \gg \sigma_1 = 0.56|H\rangle + 0.44|T\rangle = c_2 \gg \sigma_2 \\
\text{for: } \\
\sigma_1 = \frac{1}{3}|1\rangle + \frac{2}{3}|0\rangle \quad \sigma_2 = \frac{2}{3}|1\rangle + \frac{1}{3}|0\rangle \\
\text{and } \sigma_1 = \frac{3}{8}|H\rangle + \frac{5}{8}|T\rangle \quad \sigma_2 = \frac{1}{8}|H\rangle + \frac{7}{8}|T\rangle .

We note that $\begin{array}{c}
\hline \\
\hline \\
\end{array} \begin{array}{c}
\hline \\
\hline \\
\end{array} \begin{array}{c}
\hline \\
\hline \\
\end{array}$ is not an accessible string diagram: the wire inbetween the two boxes cannot be accessed from the outside. If these wires are accessible, then we can access the individual boxes of a string diagram and use disintegration to compute them. We illustrate how this works.
Example 3.9.1. Consider two-element sets $A = \{a, a'\}$, $B = \{b, b'\}$, $C = \{c, c'\}$ and $D = \{d, d'\}$ and an (accessible) string diagram $S$ of the form:

$$S = \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
A
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
C
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
D
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
B
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\sigma
\end{array}
\end{array}
\end{array}
\end{array}
\end{array} \quad (3.27)$$

Now suppose we have a joint state $\omega \in \mathcal{D}(A \times C \times D \times B)$ given by:

$$\omega = 0.04|a, c, d, b\rangle + 0.18|a, c, d, b'\rangle + 0.06|a, c, d', b\rangle + 0.02|a, c, d', b'\rangle$$

$$+ 0.04|a, c', d, b\rangle + 0.18|a, c', d, b'\rangle + 0.06|a, c', d', b\rangle$$

$$+ 0.02|a, c', d', b'\rangle + 0.024|a', c, d, b\rangle + 0.018|a', c, d, b'\rangle$$

$$+ 0.036|a', c, d', b\rangle + 0.002|a', c, d', b'\rangle + 0.096|a', c', d, b\rangle$$

$$+ 0.072|a', c', d, b'\rangle + 0.144|a', c', d', b\rangle + 0.008|a', c', d', b'\rangle$$

We ask ourselves: does $\omega \models S$ hold? More specifically, can we somehow obtain interpretations of the boxes $\sigma$, $f$ and $g$ in $S$ so that $\omega = \ll S \gg$? We shall show that by appropriately using marginalisation and disintegration we can ‘factorise’ this joint state according to the above string diagram.

First we can obtain the state $\sigma \in \mathcal{D}(A \times B)$ by discarding the $C, D$ outputs in the middle, as in:

$$\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\sigma
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\sigma
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\sigma
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\sigma
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array} = \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\sigma
\end{array}
\end{array}
\end{array} \quad = \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\sigma
\end{array}
\end{array}
\end{array} \quad = \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\sigma
\end{array}
\end{array}
\end{array}$$

We can thus compute $\sigma$ as:

$$\sigma = \omega[1, 0, 0, 1]$$

$$= (\sum_{u,v} \omega(a, u, v, b))|a, b\rangle + (\sum_{u,v} \omega(a, u, v, b'))|a, b'\rangle$$

$$+ (\sum_{a', u} \omega(a', u, v, b))|a', b\rangle + (\sum_{a', u} \omega(a', u, v, b'))|a', b'\rangle$$

$$= (0.04 + 0.06 + 0.04 + 0.06)|a, b\rangle$$

$$+ (0.18 + 0.02 + 0.18 + 0.02)|a, b'\rangle$$

$$+ (0.024 + 0.036 + 0.096 + 0.144)|a', b\rangle$$

$$+ (0.018 + 0.002 + 0.072 + 0.008)|a', b'\rangle$$

$$= 0.2|a, b\rangle + 0.4|a, b'\rangle + 0.3|a', b\rangle + 0.1|a', b'\rangle.$$
3.9. Factorisation of joint states

We first illustrate how to restrict the string diagram to the relevant part via marginalisation. For \( f \) we concentrate on:

\[
\begin{aligned}
\sigma \quad \sigma
\end{aligned}
\]

The string diagram on the right tells us that we can obtain \( \omega \) via disintegration from the marginal \( \omega[1, 1, 0] \), using that extracted channels are unique, in diagrams of this form, see \[\text{(3.18)}.\] In the same way one obtains \( g \) from the marginal \( \omega[0, 0, 1] \). Below we directly give the outcomes, and leave the details of the calculations to the reader.

\[
\begin{aligned}
f &= \omega[0, 1, 0, 0][1, 0, 0, 0] \\
&= \begin{cases} 
\sigma \quad \sigma \\
a \mapsto \frac{\sum_{c,v} \omega(a, c, v, y)}{\sum_{u,v} \omega(a, u, v, y)} |c\rangle + \frac{\sum_{c,v} \omega(a, c^+, v, y)}{\sum_{u,v} \omega(a, u, v, y)} |c^+\rangle \\
a^+ \mapsto 0.5 |c\rangle + 0.5 |c^+\rangle \\
a^- \mapsto 0.2 |c\rangle + 0.8 |c^+\rangle 
\end{cases} \\
g &= \omega[0, 0, 1, 0][0, 0, 0, 1] \\
&= \begin{cases} 
\sigma \quad \sigma \\
b \mapsto \frac{\sum_{x,d,b} \omega(x, u, d, b)}{\sum_{x,v,b} \omega(x, u, v, b)} |d\rangle + \frac{\sum_{x,d,b} \omega(x, u, d^+, b)}{\sum_{x,v,b} \omega(x, u, v, b)} |d^+\rangle \\
b^+ \mapsto 0.4 |d\rangle + 0.6 |d^+\rangle \\
b^- \mapsto 0.9 |d\rangle + 0.1 |d^+\rangle 
\end{cases}
\end{aligned}
\]

At this stage one can check that the joint state \( \omega \) can be reconstructed from these extracted state and channels, namely as:

\[
\| S \| = (\text{id} \otimes f \otimes g \otimes \text{id}) \gg (\Delta \otimes \Delta) \gg \sigma = \omega.
\]

This proves \( \omega \models S \).

We now come to the definition of \( \models \). We will use it for channels, and not just for states, as used above.

**Definition 3.9.2.** Let \( A_1, \ldots, A_n, B_1, \ldots, B_m \) be finite sets. Consider a channel \( c: A_1 \times \cdots \times A_n \rightarrow B_1 \times \cdots \times B_m \) and a string diagram \( S \in SD(\Sigma) \) over signature
Σ, with domain $[A_1, \ldots, A_n]$ and codomain $[B_1, \ldots, B_m]$. We say that $c$ and $S$ have the same type.

In this situation we write:

$$c \models S$$

if there is an interpretation of the boxes in $\Sigma$ such that $c = [S]$.

We then say that $c$ has shape $S$ and also that $c$ factorises according to $S$. Alternative terminology is: $c$ is Markov relative to $S$, or: $S$ represents $c$. Given $c$ and $S$, finding an interpretation of $\Sigma$ such that $c = [S]$ is called a factorisation problem. It may have no, exactly one, or more than one solution (interpretation), as we have seen above.

The following illustration is a classical one, showing how seemingly different shapes are related. It is often used to describe conditional independence — in this case of $A, C$, given $B$.

**Lemma 3.9.3.** Let a state $\omega \in D(\Lambda \times B \times C)$ have full support. Then:

$$\omega \models \begin{array}{c} c \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \diamond \end{array} \iff \omega \models \begin{array}{c} c \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \diamond \end{array} \iff \omega \models \begin{array}{c} c \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \diamond \end{array}$$

The string diagrams on the left is often called a fork; the other two are called chain.

**Proof.** Since $\omega$ has full support, so have all its marginals. This allows us to perform all disintegrations below. We start on the left-hand-side, and assume an interpretation $\omega = \langle c, \text{id}, d \rangle \Rightarrow \tau$, consisting of a state $\tau = \omega[0, 1, 0] \in D(B)$ and channels $c : B \rightarrow A$ and $d : B \rightarrow C$. We write $\sigma = c \Rightarrow \tau = \omega[1, 0, 0]$ and take the Bayesian inversion $c^\dagger : A \rightarrow B$. We now have an interpretation of the string diagram in the middle, which is equal to $\omega$, since by (3.21):

$$\begin{array}{c} c \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \diamond \end{array} = \begin{array}{c} c \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \diamond \end{array} \iff \begin{array}{c} c \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \diamond \end{array}$$

Similarly one obtains an interpretation of the string diagram on the right via $\rho = d \Rightarrow \tau = \omega[0, 0, 1]$ and the inversion $d^\dagger : C \rightarrow B$. 
3.9. Factorisation of joint states

In the direction (⇐) one uses Bayesian inversion in a similar manner to transform one interpretation into another one.

In Example 3.7.5 we have extracted certain structure from a joint state, given a particular string diagram (or shape) (3.24). This can be done quite generally, essentially as in Example 3.9.1. But note that the resulting interpretation of the string diagram need not be equal to the original joint state — as Example 3.7.5 shows.

**Proposition 3.9.4.** Let $c$ be a channel with full support, of the same type as an accessible string diagram $S \in SD(\Sigma)$ that does not contain $\top$. Then there is a unique interpretation of $\Sigma$ that can be obtained from $c$. In this way we can factorise $c$ according to $S$.

In the special case that $c \models S$ holds, the factorisation interpretation of $S$ obtained in this way from $c$ is the same one that gives $\llbracket S \rrbracket = c$ because of $c \models S$ and uniqueness of disintegrations.

**Proof.** We conveniently write multiple wires as a single wire, using product types. We use induction on the number of boxes in $S$. If this number is zero, then $S$ only consists of 'structural' elements, and can be interpreted irrespective of the channel $c$.

Now let $S$ contain at least one box. Pick a box $g$ at the top of $S$, with all of its output wires directly coming out of $S$. Thus, since $S$ is accessible, we may assume that it has the form as described on the left below.

\[
S = g \quad \text{so that} \quad S' := \quad \quad = \\
\quad \quad h
\]

By discarding the wire on the left we get a diagram $S'$ as used in disintegration, see Definition 3.5.4. But then a unique channel (interpretation) $g$ can be extracted from $\llbracket S' \rrbracket = c[0, 1, 1]$.

We can now apply the induction hypothesis with the channel $c[1, 1, 0]$ and corresponding string diagram from with the box $g$ has been removed. It will give an interpretation of all the other boxes — not equal to $g$ — in $S$.

This result gives a way of testing whether a channel $c$ has shape $S$: factorise $c$ according to $S$ and check if the resulting interpretation $\llbracket S \rrbracket$ equals $c$. Unfortunately, this is computationally rather expensive.
3.9.1 Shapes under conditioning

We have seen that a distribution can have a certain shape. An interesting question that arises is: what happens to such a shape when the distribution is updated? The result below answers this question, much like in [54], for three basic shapes, called fork, chain and collider.

**Proposition 3.9.5.** Let $\omega \in \mathcal{D}(X \times Y \times Z)$ be an arbitrary distribution and let $q \in \text{Fact}(Y)$ be a factor on its middle component $Y$. We write $a \in Y$ for an arbitrary element with associated point predicate $1_a$.

1. Let $\omega$ have fork shape:

$$
\omega \models \begin{array}{c}
\square \\
\square
\end{array}
\text{ then also } \omega|_{1\otimes\sigma} \models \begin{array}{c}
\square \\
\square
\end{array}.
$$

In the special case of conditioning with a point predicate we get:

$$
\omega|_{1\otimes1_a} \models \begin{array}{c}
\square \\
\square
\end{array}.
$$

2. Suppose $\omega$ has chain shape

$$
\omega \models \begin{array}{c}
\square \\
\square
\end{array}
\text{ then also } \omega|_{1\otimes\sigma} \models \begin{array}{c}
\square \\
\square
\end{array}.
$$

In the special case of a point predicate we get:

$$
\omega|_{1\otimes1_a} \models \begin{array}{c}
\square \\
\square
\end{array}.
$$

3. Let $\omega$ have collider shape:

$$
\omega \models \begin{array}{c}
\square \\
\square
\end{array}
\text{ then } \omega|_{1\otimes\sigma} \models \begin{array}{c}
\square \\
\square
\end{array}.
$$

For this shape it does not matter if $q$ is a point predicate or not.

**Proof.** Let’s assume we have an interpretation $\omega = (c, \text{id}, d) \gg \sigma$, for $c \colon Y \rightarrow X, d \colon Y \rightarrow Z$ and $\sigma \in \mathcal{D}(Y)$. Then:

$$
\omega|_{1\otimes\sigma} = ((c, \text{id}, d) \gg \sigma)|_{1\otimes\sigma} = \langle c, \text{id}, d \rangle|_{1\otimes\sigma} \gg (\sigma)|_{1\otimes\sigma} \gg \text{ by Corollary 2.3.8(2)}
$$

Hence we see the same shape that $\omega$ has.
3.9. Factorisation of joint states

In the special case when $q = 1_a$ we can extend the above calculation and obtain a parallel product of states:

$$\omega|_{1_a \otimes 1} = (c, id, d) \gg (\sigma|_a)$$

as just shown

$$= ((c \otimes id \otimes d) \gg \Delta_1) \gg 1|a)$$

by Lemma 2.3.5 (2)

$$= (c \otimes id \otimes d) \gg (\Delta_1 \gg 1|a))$$

$$= (c \otimes id \otimes d) \gg (1|a) \otimes (1|a) \otimes (d \gg 1|a))$$

Thus we see the same chain shape again.

We now write the distribution $\omega$ via its interpretation of the chain shape as:

$$\omega = (id \otimes (id, d)) \gg ((id, c) \gg \sigma) = (id, (id, d) \gg c) \gg \sigma,$$

for $\sigma \in \mathcal{D}(X)$ with $c : X \rightarrow Y$ and $d : Y \rightarrow Z$. Then:

$$\omega|_{1_a \otimes 1} = (((id, (id, d) \gg c) \gg \sigma)|_{1_a \otimes 1}$$

by Corollary 2.5.8 (2)

$$= (id, (id, d) \gg c)|_{1_a \otimes 1} \gg (\sigma|_{id \otimes (id, d) \gg c \otimes (id, d) \gg c})$$

by Corollary 2.5.8 (1)

Thus we see the same chain shape again.

When $q = 1_a$ we get a parallel product of three states:

$$\omega|_{1_a \otimes 1}(x, y, z) = (\frac{(id, (id, d) \gg c) \gg \sigma)(x, y, z) \cdot (1 \otimes 1_a \otimes 1)(x, y, z)}{(id, (id, d) \gg c) \gg \sigma \models 1 \otimes 1_a \otimes 1})$$

3 Let’s assume as interpretation of the collider shape:

$$\omega = (id \otimes c \otimes id) \gg ((\Delta \otimes \Delta) \gg (\sigma \otimes \tau)).$$
Chapter 3. Directed Graphical Models

for states $\sigma \in \mathcal{D}(X)$ and $\tau \in \mathcal{D}(Z)$ and channel $c: X \times Z \to Y$. Then:

$$\omega\mid_{1 \otimes q \otimes 1} = \left((\text{id } \otimes c \otimes \text{id}) \gg ((\Delta \otimes \Delta) \gg (\sigma \otimes \tau))\right)\mid_{1 \otimes q \otimes 1}$$

by Corollary 2.5.8 (3)

$$\omega\mid_{1 \otimes q \otimes 1} = \left((\text{id } \otimes c \mid_q \otimes \text{id}) \gg ((\Delta \otimes \Delta) \gg (\sigma \otimes \tau))\mid_{1 \otimes (c \ll q) \otimes 1}\right)$$

by Theorem 2.5.7

The updated state $((\sigma \otimes \tau)\mid_{(\Delta \otimes \Delta) \ll (1 \otimes (c \ll q) \otimes 1)})$ is typically entwined, even if $q$ is a point predicate, see also Exercise 2.3.6.

The fact that conditioning with a point predicate destroys the shape, in the above first two points, is an important phenomenon since it allows us to break entwinedness / correlations. This is relevant in statistical analysis, esp. w.r.t. causality [83, 84], see Section ??.

Exercises

3.9.1 Check the aim of Exercise 2.8.1 really is to prove

$$\omega \models \square$$

for the (ternary) state $\omega$ defined there.

3.9.2 Let $S \in SD(\Sigma)$ be given with an interpretation of the string diagram signature $\Sigma$. Check that then $\llbracket S \rrbracket \models S$.

3.10 Inference in Bayesian networks, reconsidered

Inference is one of the main topics in Bayesian probability. We have seen illustrations of Bayesian inference in Section 2.7 for the Asia Bayesian network, using forward and back inference along a channel (see Definition 2.5.1). We
are now in a position to approach the topic of inference more systematically, using string diagrams. We start by defining inference itself, namely as what we have earlier called crossover inference.

Let \( \omega \in D(X_1 \times \cdots \times X_n) \) be a joint state on a product space \( X_1 \times \cdots \times X_n \). Suppose we have evidence \( p \) on one component \( X_i \), in the form of a factor \( p \in \text{Fact}(X_i) \). After updating with \( p \), we are interested in the marginal distribution at component \( j \). We will assume that \( i \neq j \), otherwise the problem is simple and can be reduced to updating the marginal at \( i = j \) with \( p \), see Lemma [2.3.5](6).

In order to update \( \omega \) on \( X_1 \times \cdots \times X_n \) with \( p \in \text{Fact}(X_i) \) we first have to extend the factor \( p \) to a factor on the whole product space, by weakening it to:

\[
\pi_i \ll p = 1 \otimes \cdots \otimes 1 \otimes p \otimes 1 \otimes \cdots \otimes 1.
\]

After the update with this factor we take the marginal at \( j \) in the form of:

\[
\pi_j \gg (\omega| \pi_i \ll p) = \omega|_i \ll p[0, \ldots, 0, 1, 0, \ldots, 0].
\]

The latter provides a convenient form.

**Definition 3.10.1.** A basic inference query is of the form:

\[
\pi_j \gg (\omega|_i \ll p)
\]

for a joint state \( \omega \) with an evidence factor \( p \) on its \( i \)-th component and with the \( j \)-th marginal as conclusion.

What is called inference is the activity of calculating the outcome of an inference query.

An inference query may also have multiple evidence factors \( p_k \in \text{Fact}(X_{i_k}) \), giving an inference query of the form:

\[
\pi_j \gg (\omega|_{i_1 \ll p_1} \otimes \cdots \otimes |_{i_m \ll p_m}) = \pi_j \gg (\omega|_{i_1 \ll p_1} \cdots |_{i_m \ll p_m}).
\]

By suitably swapping components and using products one can reduce such a query to one of the form (3.28). Similarly one may marginalise on several components at the same time and again reduce this to the canonical form (3.28).

The notion of inference that we use is formulated in terms of joint states. This gives mathematical clarity, but not a practical method to actually compute queries. If the joint state has the shape of a Bayesian network, we can use this network structure to guide the computations. This is formalised in the next result: it described quite generally what we have been illustrating many times already, namely that inference can be done along channels — both forward and backward — in particular along channels that interprete edges in a Bayesian network.
Theorem 3.10.2. Let joint state $\omega$ have the shape $S$ of a Bayesian network: $\omega \models S$. An inference query $\pi_j \gg (\omega|_{\tau<i}\ll p)$ can then be computed via forward and backward inference along channels in the string diagram $S$.

Proof. We use induction on the number of boxes in $S$. If this number is one, $S$ consists of a single state box, whose interpretation is $\omega$. Then we are done by the crossover inference Corollary 2.5.9.

We now assume that $S$ contains more than one box. We pick a box at the top of $S$ so that we can write:

$$\omega = (\text{id}_{X_1 \times \cdots \times X_{i-1}} \otimes c \otimes \text{id}_{X_{i+1} \times \cdots \times X_n}) \gg \rho \quad \text{where} \quad \rho \models S'$$

and where $S'$ is the string diagram obtained from $S$ by removing the single box whose interpretation we have written as channel $c$, say of type $Y \rightarrow X_k$.

The induction hypothesis applies to $\rho$ and $S'$.

Consider the situation below wrt. the underlying product space $X_1 \times \cdots \times X_n$ of $\omega$,

\[
\begin{array}{c}
\text{evidence} \quad \rho \\
\downarrow \quad \downarrow \quad \downarrow \\
X_1 \times \cdots \times X_i \times \cdots \times X_k \times \cdots \times X_j \times \cdots \times X_n \\
\uparrow c
\end{array}
\]

Below we distinguish three cases about (in)equality of $i, j, k$ where, recall, that we assume $i \neq k$.

1. $i \neq k$ and $j \neq k$. In that case we can compute the inference query as:

$$\pi_j \gg (\omega|_{\tau<i}\ll p) = \pi_j \gg (\text{id} \otimes c \otimes \text{id}) \gg \rho|_{\tau<i}\ll p)$$

$$= \pi_j \gg (\text{id} \otimes c \otimes \text{id}) \gg (\rho|_{\tau<i}\ll p) \quad \text{by Exercise 2.3.5(1)}$$

$$= \pi_j \gg (\rho|_{\tau<i}\ll p).$$

Hence we are done by the induction hypothesis.

Aside: we are oversimplifying by assuming that the domain of the channel $c$ is not a product space; if it is, say $Y_1 \times \cdots \times Y_m$ of length $m > 1$ we should not marginalise at $j$ but at $j + m - 1$. A similar shift may be needed for the weakening $\pi_i \ll p$ if $i < k$.

2. $i = k$, and hence $j \neq k$. Then:

$$\pi_j \gg (\omega|_{\tau<i}\ll p) = \pi_j \gg (\rho|_{\tau<i}\ll (\text{id} \otimes c \otimes (\text{id} \otimes c \ll p)))$$

$$= \pi_j \gg (\rho|_{\tau<i}\ll (\rho|_{\tau<i}\ll p)) \quad \text{by Exercise 2.3.5(2)}$$

The induction hypothesis now applies with transformed evidence $c \ll p$. 

196
3.10. Inference in Bayesian networks, reconsidered

Figure 3.2 Two ‘stretchings’ of the Asia Bayesian network from Section 2.7, with the same semantics.

Aside: again we are oversimplifying; when the channel \( c \) has a product space as domain, the single projection \( \pi_i \) in the conditioning-factor \( \pi_i \ll (c \ll p) \) may have to be replaced by an appropriate tuple of projections.

\[
j = k, \text{ and hence } i \neq k. \text{ Then:}
\]
\[
\pi_j \gg (\omega|_{x_j \ll p}) = \pi_j \gg (\rho|_{x_j \ll p})
\]
\[
= \pi_j \gg (\rho|_{x_j \ll p}) \quad \text{by Exercise 2.3.5(1)}
\]
\[
= c \gg (\rho|_{x_j \ll p}).
\]

Again we are done by the induction hypothesis.

Instead of a channel \( c \), as used above, one can have a diagonal or a swap map in the string diagram \( S \). Since diagonals and swaps are given by functions \( f \) — forming trivial channels — Lemma 2.3.5(7) applies, so that we can proceed via evaluation of \( \pi_i \circ f \).

Example 3.10.3. We shall recompute the query ‘smoking, given a positive xray’ in the Asia Bayesian network from Section 2.7, this time using the mechanism of Theorem 3.10.2. In order to do this we have to choose a particular order in the channels of the network. Figure 3.2 give two such orders, which we also call ‘stretchings’ of the network. These two string diagrams have the
same semantics, so it does not matter which one we take. For this illustration we choose the one on the left.

This means that we can write the joint state $\omega \in D(S \times D \times X)$ as composite:

$$\omega = (id \otimes dysp \otimes id) \circ (id \otimes id \otimes id \otimes xray) \circ (id \otimes id \otimes \Delta_2)$$

$$\circ (id \otimes id \otimes either) \circ (id \otimes id \otimes id \otimes tub) \circ (id \otimes id \otimes id \otimes asia)$$

$$\circ (id \otimes id \otimes lung) \circ (id \otimes dysp \otimes id) \circ \Delta_3 \circ smoking$$

The positive x-ray evidence translates into a point predicate $1$, on the set $X = \{x, x'\}$ of x-ray outcomes. It has to be weakened to $\pi_1 \ll 1 = 1 \otimes 1 \otimes 1$, on the product space $S \times D \times X$. We are interested in the updated smoking distribution, which can be obtained as first marginal. Thus we compute the following inference query, where the number $k$ in $\pi$ refers to the $k$-th of the three distinctions in the proof of Theorem 3.10.2.

$$\pi_1 \gg (\omega |_{xray=1})$$

$$= \pi_1 \gg ((id \otimes dysp \otimes id) \circ \cdots |_{xray=1})$$

$$= \pi_1 \gg ((id \otimes id \otimes xray) \circ \cdots |_{xray=1})$$

$$= \pi_1 \gg ((id \otimes id \otimes \Delta_2) \circ \cdots |_{xray=1})$$

$$= \pi_1 \gg ((id \otimes id \otimes either) \circ \cdots |_{xray=1})$$

$$= \pi_1 \gg ((id \otimes id \otimes tube) \circ \cdots |_{xray=1})$$

$$= \pi_1 \gg ((id \otimes id \otimes asia) \circ \cdots |_{xray=1})$$

$$= \pi_1 \gg ((id \otimes dysp \otimes id) \circ \cdots |_{xray=1})$$

$$= \pi_1 \gg ((id \otimes id \otimes asia) \circ \cdots |_{xray=1})$$

$$= \pi_1 \gg ((id \otimes dysp \otimes id) \circ \cdots |_{xray=1})$$

$$= \pi_1 \gg ((\Delta_3 \circ smoking) |_{xray=1})$$

$$= smoking |_{xray=either(id \otimes tube)(id \otimes xray) \circ smoking} = 1$$

$$= 0.6878|s(x) + 0.3122|x'\rangle.$$

It is a exercise below to compute the same inference query for the stretching on the right in Figure 3.2, since there it is shown that all such inference computations are equal to the computation on the joint state.

The inference calculation in Example 3.10.3 is quite mechanical in nature and can thus be implemented easily, giving a channel-based inference algorithm, see [46] for Bayesian networks. The algorithm consists of two parts.

1. It first finds a stretching of the Bayesian network with a minimal \textit{width}, that
3.11. Updating and adapting: Pearl versus Jeffrey

... is, a description of the network as a sequence of channels such that the state space in between the channels has minimal size. As can be seen in Figure 3.2, there can be quite a bit of freedom in choosing the order of channels.

2 Perform the calculation as in Example 3.10.3 following the steps in the proof of Theorem 3.10.2.

The resulting algorithm’s performance compares favourably to the performance of the pgmpy Python library for Bayesian networks [3]. By design in uses (fuzzy) factors and not (sharp) events and thus solves the “soft evidential update” problem [99].

Exercises

3.10.1 In Example 3.10.3 we have identified a state on a space $X$ with a channel $1 \rightarrow X$, as we have done before, e.g. in Exercise 1.7.2. Consider states $\sigma \in \mathcal{D}(X)$ and $\tau \in \mathcal{D}(Y)$ with a factor $p \in \mathsf{Fact}(X \times Y)$. Prove that, under this identification,

$$\sigma_{\mid\left(id \otimes \tau\right) \triangleleft p} = (\sigma \otimes \tau)_{\mid p}[1, 0].$$

3.10.2 Compute the inference query from Example 3.10.3 for the stretching of the Asia Bayesian network on the right in Figure 3.2.

3.10.3 From Theorem 3.10.2 one can conclude that the for the channel-based calculation of inference queries the particular stretching of a Bayesian network does not matter — since they are all equal to inference on the joint state. Implicitly, there is a ‘shift’ result about forward and backward inference.

Let $c: A \rightarrow X$ and $d: B \rightarrow Y$ be channels, together with a joint state $\omega \in \mathcal{D}(A \times X)$ and a factor $q: Y \rightarrow \mathbb{R}_{\geq 0}$ on $Y$. Then the following marginal distributions are the same.

$$\left(\left(c \otimes id\right) \gg \omega\right)_{\mid (id \otimes d) \triangleleft \omega}[1, 0] = \left(\left(c \otimes id\right) \gg \left((id \otimes d) \gg \omega\right)_{\mid 1, 0}\right)[1, 0].$$

1 Prove this equation.

2 Give an interpretation of this equation in relation to inference calculations.

3.11 Updating and adapting: Pearl versus Jeffrey

The evidence that is standardly used in Bayesian updating is in the form of events, that is, of subsets, or equivalently, of sharp predicates. In this book, in
contrast, we have been using fuzzy evidence right from the start, with values in the unit interval $[0, 1]$, or more generally, in $\mathbb{R}_{\geq 0}$. Updating has been defined for such fuzzy evidence, see Section 2.3.

In the Bayesian literature however this approach is non-standard, and updating with fuzzy (or: soft or uncertain) evidence has been seen as problematic. Two approaches emerged, one named after Pearl [80, 82], and one after Jeffrey [56], see e.g. [11, 20, 22, 99].

Pearl’s approach is described operationally: extend a Bayesian network with an auxiliary binary node, so that soft evidence can be emulated in terms of hard evidence on this additional node, and so that the usual inference methods can be applied. This extension of a Bayesian network at node $X$ with a binary node 2 and an associated conditional probability table, corresponds to adding a channel $X \to 2$. But such channels correspond to fuzzy predicates $X \to [0, 1]$, since $\mathcal{D}(2) \equiv [0, 1]$, see also Exercise 2.4.8.

What is called Pearl’s rule can be identified with what we call backward inference, in Section 2.5. We have seen that backward inference forms a powerful reasoning method, where we have a channel $c: X \to Y$ that mediates between a state $\omega \in \mathcal{D}(X)$ on the channel’s domain $X$, that must be updated with evidence $q \in \text{Fact}(Y)$ on the channel’s codomain $Y$. This works via the formula $\omega|_c \ll q \in \mathcal{D}(X)$, where the factor $q$ is first transformed along the channel $c$ and then used for updating the state $\omega$.

Jeffrey’s rule forms an alternative, for which we shall use the word ‘adaptation’ instead of ‘update’. It applies in a similar situation, with a channel $c: X \to Y$ and a state $\omega \in \mathcal{D}(X)$. But now there is no evidence factor on $Y$, but a state $\rho \in \mathcal{D}(Y)$. It turns out that there is snappy description of Jeffrey’s rule using the dagger (Bayesian inversion) $c^\dagger$ of the channel $c$ and state transformation along $c^\dagger$, see [49]. One then adapts $\omega$ to a new state, namely: $c^\dagger_\omega \gg \rho$.

We summarise the situation.

**Definition 3.11.1.** Let $c: X \to Y$ be channel with a prior state $\omega \in \mathcal{D}(X)$.

1 *Pearl’s rule* involves evidence-based *updating* of the prior $\omega$ with a factor $q \in \text{Fact}(Y)$ to the posterior:

$$\omega|_c \ll q \in \mathcal{D}(X).$$

2 *Jeffrey’s rule* involves state-based *adaptation* of the prior $\omega$ with a state $\rho \in \mathcal{D}(Y)$ to the posterior:

$$c^\dagger_\omega \gg \rho \in \mathcal{D}(X).$$

We shall illustrate the difference between Pearl’s and Jeffreys’s rules.
3.11. Updating and adapting: Pearl versus Jeffrey

Example 3.11.2. Consider the Bayesian network / string diagram below, with its interpretation via two states and a channel \( \text{alarm} : B \times E \rightarrow A \) as defined on the right, for 2-element sets \( B = \{ b, b^\perp \} \), \( E = \{ e, e^\perp \} \) and \( A = \{ a, a^\perp \} \).

\[
\begin{array}{c}
B \\
\text{burglary}
\end{array}
\begin{array}{c}
E \\
\text{earthquake}
\end{array}
\rightarrow
\begin{array}{c}
A \\
\text{alarm}
\end{array}
\]

- \( \text{burglary} = 0.01|b\rangle + 0.99|b^\perp\rangle \)
- \( \text{earthquake} = 0.000001|e\rangle + 0.999999|e^\perp\rangle \)
- \( \text{alarm}(b, e) = 0.99999(a) + 0.0001(a^\perp) \)
- \( \text{alarm}(b^\perp, e) = 0.99(a) + 0.01(a^\perp) \)
- \( \text{alarm}(b^\perp, e^\perp) = 0.0001(a) + 0.9999(a^\perp) \)

The following question is asked in [5, Example 3.1 and 3.2]:

Imagine that we are 70% sure we heard the alarm sounding. What is the probability of a burglary?

Via the backward inference approach that we have used so far, that is, via Pearl’s rule, we interpret the given evidence as a fuzzy/soft predicate \( p : A \rightarrow [0, 1] \) with \( p(a) = 0.7 \) and \( p(a^\perp) = 0.3 \). This predicate \( p \) can be seen as an extension of the above Bayesian network with a box and table/interpretation:

\[
\begin{array}{c}
A \\
\text{alarm}
\end{array}
\begin{array}{c}
2
\end{array}
\]

\[
\begin{cases}
  p(a) = 0.7 \\
  p(a^\perp) = 0.3.
\end{cases}
\]

The required outcome is then obtained via backward inference, followed by marginalisation to the burglary case:

\[
(burglary \otimes earthquake)|_{\text{alarm} = p}[1, 0] = 0.0229|b\rangle + 0.9771|b^\perp\rangle.
\]

In contrast, the outcome given in [5] is:

\[
0.693|b\rangle + 0.307|b^\perp\rangle.
\]

The difference in outcomes is substantial: 2% versus 70%. What is going on?

The second answer is motivated in [5] via ‘Jeffrey’s’ rule. As mentioned above, this method can be formulated in terms of Bayesian inversion (dagger). First, we interpret the evidence as a state \( \rho = 0.7|a\rangle + 0.3|a^\perp\rangle \). We can take the Bayesian inversion of the channel \( \text{alarm} : B \times E \rightarrow A \) wrt. the product state given by \( \text{burglary} \) and \( \text{earthquake} \), and then do state transformation and marginalisation:

\[
(\text{alarm}_{\text{burglary} \otimes \text{earthquake}} \triangleright \rho)[1, 0] = 0.693|b\rangle + 0.307|b^\perp\rangle.
\]

This is the formulation of Jeffrey’s rule in Definition 3.11.1.
We look a bit closer at the difference between these two rules in this example. If we are 100% certain that we heard the alarm the burglary distribution becomes:

\[
(burglary \otimes earthquake)_{alarm \leftarrow 1, [1,0]} = 0.99|b\rangle + 0.01|b^\perp\rangle \\
= (alarm_{burglary \leftarrow earthquake} \gg 1|a\rangle)[1,0].
\]

Indeed, as we shall see below, Pearl’s and Jeffrey’s rules agree on point evidence/states.

We next notice that the outcomes of Jeffrey’s rule look linear: about 70% burglar probability for 70% certainty about the alarm, and about 100% for 100% certainty. This is indeed what the blue straight line for Jeffrey’s rule (JR) shows in Figure 3.3. It describes the function \([0,1] \rightarrow [0,1]\) given by:

\[
r \mapsto (alarm_{burglary \leftarrow earthquake} \gg (r|a\rangle + (1-r)|a^\perp\rangle))[1,0](b).
\]

The red bent line describes Pearl’s rule (PR) via the function:

\[
r \mapsto (burglary \otimes earthquake)_{alarm \leftarrow (rI_{1} + (1-r)I_{-1})}[1,0](b)
\]

The plots in Figure 3.3 and 3.4 show that changing the prior burglary probability gives different outcomes, but does not fundamentally change the shapes.

The question remains: is there a ‘right’ way to do update/adapt in this situation, via Pearl’s or via Jeffrey’s rule, and if so, which one is right, under
which circumstances? This is an urgent question, since important decisions, e.g. about medical treatments, are based on probabilistic inference.

We briefly focus on some commonalities and differences.

**Lemma 3.11.3.** Let \( c : X \rightarrow Y \) be a channel with a prior state \( \omega \in \mathcal{D}(X) \).

1. Pearl's rule and Jeffrey's rule agree on points: for \( y \in Y \), with associated point predicate \( 1_y \) and point state \( 1|y \rangle \), one has:
   \[
   \omega|_{c \ll 1_y} = c_{\omega}^\dagger(y) = c_{\omega}^\dagger \gg 1|y \rangle.
   \]

2. Pearl's updating with a constant predicate (no information) does not change the prior state \( \omega \):
   \[
   \omega|_{c \ll r \cdot 1} = \omega, \quad \text{for } r > 0.
   \]

3. Jeffrey's adaptation with the predicted state \( c \gg \omega \) does not change the prior state \( \omega \):
   \[
   c_{\omega}^\dagger \gg (c \gg \omega) = \omega.
   \]

4. Multiple updates with Pearl's rule commute, but multiple adaptations with Jeffrey's rule do not commute.

**Proof.** These points follow from earlier results.

1. Easy, by Proposition 3.7.2.
2 By Lemma 2.3.5 (1) and (4).
3 This is Equation (3.22).
4 By Lemma 2.3.5 (3) we have:

\[ \omega_{c \sim q_1 | c \sim q_2} = \omega_{[c \sim q_1] \& [c \sim q_2]} = \omega_{c \sim q_1 | c \sim q_1}. \]

However, in general,

\[ c^t_{c \sim q_1} \gg q_2 \neq c^t_{c \sim q_2} \gg q_1. \]

Probabilistic updating may be used as a model for what is called priming in cognition theory, see e.g. [35, 39]. It is well-known that the human mind is sensitive to the order in which information is processed, that is, to the order of priming/updating. Thus, point (4) above suggests that Jeffrey’s rule might be more appropriate in such a setting.

Points (2) and (3) present different views on ‘learning’, where learning is now used in an informal sense: point (2) says that according to Pearl’s rule you learning nothing when you get no information; but point (3) tells that according to Jeffrey you learn nothing when you are presented with what you already know. Both interpretations make sense.

This last fact about Jeffrey’s rule has led to the view that one should apply Jeffrey’s adaptation rule in situations where one is confronted with ‘surprises’ [24] or with ‘unanticipated knowledge’ [23]. This does not give a precise criterion about when to use which rule, but it does give an indication. The whole issue which is still poorly understood. We refer to [49] for more discussion on this intriguing and important matter, in terms of ‘factoring in’ evidence (Pearl) versus ‘adjusting to’ or ‘adaptation to’ evidence (Jeffrey). Here we include an example from [24] (also used in [49]) that makes this ‘surprise’ perspective more concrete.

**Example 3.11.4.** The following setting is taken from [24]. Ann must decide about hiring Bob, whose characteristics are described in terms of competence \((c \text{ or } c^\perp)\) and experience \((e \text{ or } e^\perp)\). The prior is a joint distribution on the product space \(C \times E\), for \(C = \{c, c^\perp\}\) and \(E = \{e, e^\perp\}\), given as:

\[ \omega = \frac{4}{10} |c, e| + \frac{1}{10} |c, e^\perp| + \frac{1}{10} |c^\perp, e| + \frac{4}{10} |c^\perp, e^\perp|. \]

The first marginal of \(\omega\) is the uniform distribution \(\frac{1}{2} |c| + \frac{1}{2} |c^\perp|\). It is the neutral base rate for Bob’s competence.

We use the two projection functions \(C \xrightarrow{\pi_1} C \times E \xrightarrow{\pi_2} E\) as deterministic channels along which we reason, using both Pearl’s and Jeffrey’s rules.

When Ann would learn that Bob has relevant work experience, given by point evidence \(1_e\), her strategy is to factor this in via Pearl’s rule / backward
3.11. Updating and adapting: Pearl versus Jeffrey

inference: this gives as posterior \( \omega|_{c} \cdot L \propto \omega|_{1} \cdot L \), whose first marginal is \( \frac{3}{5|c} + \frac{2}{5|c^+} \). It is then more likely that Bob is competent.

Ann reads Bob’s letter to find out if he actually has relevant experience. We quote from [24]:

Bob’s answer reveals right from the beginning that his written English is poor. Ann notices this even before figuring out what Bob says about his work experience. In response to this unforeseen learnt input, Ann lowers her probability that Bob is competent from \( \frac{1}{2} \) to \( \frac{1}{8} \). It is natural to model this as an instance of Jeffrey revision.

Bob’s poor English is a new state of affairs — a surprise — which translates to a competence state \( \rho = \frac{1}{8}|c⟩ + \frac{7}{8}|c^⊥⟩ \). Ann wants to adapt to this new surprising situation, so she uses Jeffrey’s rule, giving a new joint state:

\[
\omega' = (\pi_1^?) \gg \rho = \frac{4}{11}|c⟩ + \frac{7}{11}|c^⊥⟩.
\]

If the letter now tells that Bob has work experience, Ann will factor this in, in this new situation \( \omega' \), giving, like above, via Pearl’s rule followed by marginalisation,

\[
\omega'|_{c} \cdot L, [1, 0] = \frac{4}{11}|c⟩ + \frac{7}{11}|c^⊥⟩.
\]

The likelihood of Bob being competent is now lower than in the prior state, since \( \frac{4}{11} < \frac{1}{2} \). This example reconstructs the illustration from [24] in channel-based form, with the associated formulations of Pearl’s and Jeffrey’s rules from Definition 3.11.1 and produces exactly the same outcomes as in loc. cit.

There are translations back-and-forth between the Pearl’s and Jeffrey’s rules, due to [11]; they are translated here to the current setting.

**Proposition 3.11.5.** Let \( c: X \rightarrow Y \) be a channel with a prior state \( \omega \in \mathcal{D}(X) \) on its domain, such that \( \tau = c \gg \omega \) has full support.

1 Pearl’s updating can be expressed as Jeffrey’s adaptation, by turning a factor \( q: Y \rightarrow \mathbb{R}_{\geq 0} \) into a state \( \tau_q \in \mathcal{D}(Y) \), so that:

\[
c_\omega \gg \tau_q = \omega|_{c} \cdot q.
\]

2 Jeffrey’s adaptation can also be expressed as Pearl’s updating: for a state \( \rho \in \mathcal{D}(Y) \) write \( \rho/\tau \) for the factor \( y \mapsto \frac{\rho(y)}{\tau(y)} \); then:

\[
c_\omega \gg \rho = \omega|_{c} \cdot \rho.
\]

**Proof.** The first point is exactly Theorem 3.7.7 [1]. For the second point we first note that \( \tau \models \rho/\tau = 1 \), since \( \rho \) is a state:

\[
\tau \models \rho/\tau = \sum_y \tau(y) \cdot \frac{\rho(y)}{\tau(y)} = \sum_y \rho(y) = 1.
\]
But then, for $x \in X$,

$$
(c^\dagger_{\omega} \gg \rho)(x) = \sum_y \rho(y) \cdot c^\dagger_{\omega}(y)(x) \tag{3.23}
= \omega(x) \cdot \sum_y c(x)(y) \cdot \rho(y) / \pi(y)
= \omega(x) \cdot (c \ll \rho/\tau)(x) \quad \text{since } \tau = c \gg \omega \\
= \omega(x) \cdot (c \ll \rho/\tau)(x) \quad \text{as just shown}
= \omega(x) \cdot (c \ll \rho/\tau)(x)

$$

We have frequently seen that Pearl-style updating in one component of a joint state, as $\omega|_{1\otimes q}$, can be expressed via channels extracted from $\omega$. A natural question is whether the same can be done for Jeffrey’s approach. In order to clarify this, let’s marginalise $\omega \in \mathcal{D}(X \times Y)$ to $\sigma \in \mathcal{D}(X)$ and $\tau \in \mathcal{D}(Y)$, and disintegrate $\omega$ to $c := \omega[0,1] : X \rightarrow Y$ and $d := \omega[1,0] : Y \rightarrow X$ in:

$$
\omega = \sigma \xrightarrow{c} d = \tau
$$

As a result, $d = c^\dagger_{\omega}$ and $c = d^\dagger_{\omega}$.

In this situation we have, for a factor $q \in \text{Fact}(Y)$,

$$
\omega|_{1\otimes q} = (d, \text{id}) \gg (\tau|_q) = \sigma|_{c \ll q}.
$$

The first equation is Corollary 2.5.8 (2) and the second one is Corollary 2.5.9 (2).

A cross-over version of Jeffrey-style adaption of the joint prior state $\omega$ with $\rho \in \mathcal{D}(Y)$ is then defined as the posterior:

$$
(d, \text{id}) \gg \rho = d
$$
3.11. Updating and adapting: Pearl versus Jeffrey 207

The first marginal of this adapted joint state is \( d \gg \rho = \sigma \gg \rho \), as in Definition 3.11.1. The second marginal is the state \( \rho \) that we wish to adapt to. This explains why in the context of Jeffrey’s rule the ‘evidence’ state is often described as a posterior: \( \omega \) is adapted in such a way that its second marginal becomes equal to \( \rho \). We shall see several examples of such Jeffrey-style updating in the Exercises below, notably in 3.11.3 and 3.11.4.

Exercises

3.11.1 Check for yourself the claimed outcomes in Example 3.11.4:

1. \( \omega |\pi_{0,1} = \frac{1}{4}|c\rangle + \frac{1}{2}|c^{-}\rangle \);
2. \( \omega' := (\pi_1)_{\rho} \gg \rho = \frac{1}{14}|c, e\rangle + \frac{7}{30}|c, e^*\rangle + \frac{7}{10}|c^*, e\rangle \);
3. \( \omega'|\pi_{0,1} = \frac{1}{2}|c\rangle + \frac{7}{10}|c^*\rangle \).

3.11.2 This example is taken from [22], where it is attributed to Whitworth: there are three contenders \( A, B, C \) for winning a race with a prior distribution \( \omega = \frac{2}{11}|A\rangle + \frac{4}{11}|B\rangle + \frac{5}{11}|C\rangle \). Surprising information comes in that \( A \)’s chances have become \( \frac{1}{2} \). What are the adapted chances of \( B \) and \( C \)?

1. Split up the sample space \( X = \{A, B, C\} \) into a suitable two-element partition via a function \( f : X \to 2 \).
2. Use the uniform distribution \( \nu = \frac{1}{2}|1\rangle + \frac{1}{2}|0\rangle \) on 2 and show that Jeffrey’s rule gives as adapted distribution:

\[
\nu = \frac{1}{2} \cdot \omega|_{f|1} + \frac{1}{2} \cdot \omega|_{f|0} = \frac{1}{2}|A\rangle + \frac{2}{7}|B\rangle + \frac{5}{14}|C\rangle.
\]

3.11.3 The following alarm example is in essence due to Pearl [80], see also [20] §3.6; we have adapted the numbers in order to make the calculations a bit easier. There is an ‘alarm’ set \( A = \{a, a^{-}\} \) and a ‘burglary’ set \( B = \{b, b^{-}\} \), with the following a priori joint distribution on \( A \times B \).

\[
\omega = \frac{1}{500}|a, b\rangle + \frac{7}{500}|a, b^{-}\rangle + \frac{1}{1000}|a^{-}, b\rangle + \frac{98}{1000}|a^{-}, b^{-}\rangle.
\]

Someone reports that the alarm went off, but with only 80% certainty because of deafness.

1. Translate the alarm information into a predicate \( p : A \to [0, 1] \) and show that crossover updating leads to a burglary distribution:

\[
\omega|_{p=1} = \frac{21}{1057}|b\rangle + \frac{1036}{1057}|b^{-}\rangle.
\]
Chapter 3. Directed Graphical Models

2 Compute the extracted channel \( c = \omega[1, 0 | 0, 1] : B \to A \) via disintegration, and express the answer in the previous point in terms of backward inference using \( c \).

3 Use the Bayesian inversion/dagger \( d = c^\dagger \omega[0, 1] = \omega[0, 1 | 1, 0] : A \to B \) of this channel \( c \), see Exercise 3.7.5, to calculate the outcome of Jeffrey’s adaption as:

\[
\langle \mathrm{id}, c \rangle \gg (\frac{3}{25} | g \rangle + \frac{1}{2} | b \rangle) = \frac{19639}{47120} | b \rangle + \frac{73556}{47120} | b^+ \rangle.
\]

(Notice again the considerable difference in outcomes.)

3.11.4 The next illustration is attributed to Jeffrey, and reproduced for instance in [11, 20]. We consider three colors: green (\( g \)), blue (\( b \)) and violet (\( v \)), which are combined in a space \( C = \{g, b, v\} \). These colors apply to cloths, which can additionally be sold or not, as represented by the space \( S = \{s, s^\perp\} \). There is a prior joint distribution \( \tau \) on \( C \times S \), namely:

\[
\tau = \frac{3}{25} | g, s \rangle + \frac{9}{50} | g, s^\perp \rangle + \frac{3}{25} | b, s \rangle + \frac{9}{50} | b, s^\perp \rangle + \frac{8}{25} | v, s \rangle + \frac{2}{25} | v, s^\perp \rangle.
\]

A cloth is inspected by candlelight and the following likelihoods are reported per color: 70% certainty that it is green, 25% that it is blue, and 5% that it is violet.

1 Show that the updated sales distribution via the evidence-based (crossover) inference is:

\[
\frac{26}{50} | s \rangle + \frac{35}{50} | s^\perp \rangle.
\]

2 Prove that Jeffrey’s adaptation approach, obtained via state transformation along \( c = \tau[0, 1 | 1, 0] : C \to S \), yields:

\[
c \gg (\frac{7}{50} | g \rangle + \frac{1}{4} | b \rangle + \frac{1}{20} | v \rangle) = \frac{21}{50} | s \rangle + \frac{29}{50} | s^\perp \rangle.
\]

3 Check also that:

\[
\langle \mathrm{id}, c \rangle \gg (\frac{7}{50} | g \rangle + \frac{1}{4} | b \rangle + \frac{1}{20} | v \rangle)
\]

\[
= \frac{14}{50} | g, s \rangle + \frac{7}{25} | g, s^\perp \rangle + \frac{3}{10} | b, s \rangle
\]

\[
+ \frac{1}{20} | b, s^\perp \rangle + \frac{1}{20} | v, s \rangle + \frac{1}{100} | v, s^\perp \rangle.
\]

The latter outcome is given in [20].

3.11.5 Jeffrey’s rule is frequently formulated (notably in [36], to which we refer for details) and used (like in Exercise 3.11.2), in situations where the channel involved is deterministic. Consider an arbitrary function \( f : X \to I \), giving a partition of the set \( X \) via subsets \( U_i = f^{-1}(i) = \{x \in X | f(x) = i\} \). Let \( \omega \in \mathcal{D}(X) \) be a prior.
3.11. Updating and adapting: Pearl versus Jeffrey

1) Show that applying Jeffrey’s rule to a new state of affairs $\rho \in \mathcal{D}(I)$
gives as posterior:

$$f_\omega \gg \rho = \sum_i \rho(i) \cdot \omega_{1_{1_i}}$$
satisfying $f \gg (f_\omega \gg \rho) = \rho$.

2) Prove that:

$$d(f_\omega \gg \rho, \omega) = \bigwedge_{\omega' \in \mathcal{D}(X)} d(\omega, \omega')$$

where $d$ is the total variation distance from Section 2.8.

3.11.6) Prove that for a general, not-deterministic channel $c: X \rightarrow Y$ with
prior state $\omega \in \mathcal{D}(X)$ and state $\rho \in \mathcal{D}(Y)$ that there is an inequality:

$$d(c_\omega \gg \rho, \omega) \leq \bigwedge_{\omega' \in \mathcal{D}(X)} d(\omega, \omega') + d(c \gg \omega', \rho).$$
The aim of this chapter is to present basic structure underlying the learning methods for states and channels. We have already seen learning in a simple, frequentist form, namely as a natural transformation \( F_{lrn} : \mathcal{M} \to \mathcal{D} \) from (non-empty) multisets to probability distributions. Below we shall see that this leads to a maximal validity, in a suitable sense. In general, probabilistic learning is described as consisting of small steps that need to be repeated in order to reach a certain optimum. We concentrate on the nature of these small steps, and not on (convergence of) the repetitions: that’s a separate topic.

These small learning steps can be introduced as follows. Consider the validity expression:

\[
\omega \models p \quad \text{distribution / state} \quad \text{observable / evidence}
\]

(4.1)

As we shall understand it here, learning involves increasing this validity, by changing the state \( \omega \) into a new state \( \omega' \) such that \( \omega' \models p \geq \omega \models p \). Thus, in learning one takes the evidence \( p \) as a given, fixed datum that one needs to adjust to. Learning happens by changing the state \( \omega \) so that it better fits the evidence.

This may be understood in an amateur-psychological sense: the state \( \omega \) somehow captures (a consistent portion of) one’s state of my mind. When confronted with evidence \( p \), one learns \( p \) by changing one’s internal state \( \omega \) into \( \omega' \), where the validity of \( p \) is higher in \( \omega' \) than in \( \omega \). Thus, in learning one adapts one’s mind to the world, as given by the evidence \( p \).

One important way to learn in the above situation is to update (condition) the state \( \omega \) with the evidence \( p \), as introduced in Section 2.3. Indeed, a fundamental
4.1 Learning a channel from tabular data

result in this chapter, namely Theorem 4.3.1, tells that there is an inequality:

$$\omega|p|p\geq\omega|p.$$  

This captures an important intuition behind conditioning with \(p\): it changes the state so that \(p\) becomes ‘more true’.

Earlier we have seen validity \(\omega|p\) of an observable in a state, like in (4.1), but also validity \(H|\vec{p}\) of a list of observables \(\vec{p}\) in a hidden Markov model \(H\), see Section 3.4. In this chapter we shall define new forms of validity \(|\cdot|\) and \(|\cdot|\) for multisets of data. In each of these cases we shall develop an associated form of learning, via step-wise increase of validity, by introducing a ‘better’ state. We provide explicit proofs for all of these increases, which all go back to an elementary method from [7]. It is reproduced here as a ‘sum-increase lemma’, see Lemma 4.4.2. This method tells us that we should look for a maximum of a certain function, which is determined each time via elementary real analysis, using the Lagrange multiplier method. We shall see many instances of this powerful technique, which forms the basis for all learning results in this chapter.

These learning methods are applied to Markov chains and hidden Markov models in Section 4.5 and 4.6. In addition, they are used to describe the general Expectation-Maximisation (EM) learning method [21]. It consists of two parts, called Expectation (E) and Maximisation (M). We identify them as learning a ‘latent’ state along a channel, and learning a channel itself. We shall identify several forms of Expectation-Maximisation, for the different forms of validity \(|\cdot|\) and \(|\cdot|\) and also for their combination. Since there are so many versions of learning, this chapter elaborates many examples — mostly taken from the literature — in order to clarify the situation.

4.1 Learning a channel from tabular data

So far we have concentrated on learning a state (as empirical distribution) from data, via the frequentist learning operation \(Flrn: \mathcal{M} \rightarrow \mathcal{D}\), introduced in Section 1.6. This map \(Flrn\) is natural, which implies in particular that it commutes with marginalisation. In this first, short section we concentrate on how to learn a channel from multi-dimensional (tabular) data. There are in principle two ways of doing so:

1 turn the whole table into an empirical distribution, via \(Flrn\), and then apply disintegration to the resulting joint state in order to extract the right channel;
2 extract the appropriate multiset channel directly from the table, and apply
frequentist learning pointwise, in order to turn this $\mathcal{M}$-channel into a $\mathcal{D}$-channel.

The main result of this section is that both ways coincide. This substantiates the claim from Section 3.5 that disintegration is the analogue of extraction for powerset and multiset.

Recall from Subsection 1.4.5 that for a table, or joint multiset, $\tau \in \mathcal{M}(X \times Y)$ there is a function $\text{extr}_1(\tau): X \rightarrow \mathcal{M}(Y)$ given by: $\text{extr}_1(\tau)(x) = \sum_y \tau(x, y)$.

It describes the ‘row’ of the table $\tau$ at position $x$.

**Proposition 4.1.1.** Let’s use the ad hoc notation $\mathcal{M}_1(X \times Y) \subseteq \mathcal{M}(X \times Y)$ for the subset of multisets $\tau$ for which the first marginal has full support: for each $x \in X$ there is a $y \in Y$ with $\tau(x, y) > 0$.

Extraction for multisets and disintegration for distributions commute with frequentist learning, in the sense that the following diagram commutes.

$$
\begin{array}{ccc}
\mathcal{M}_1(X \times Y) & \xrightarrow{\text{extr}_1} & \mathcal{M}(Y)^X \\
\text{Flrn} & & \downarrow \text{Flrn}^X \\
\mathcal{D}(X \times Y) & \xrightarrow{\text{extr}_1} & \mathcal{D}(Y)^X
\end{array}
$$

where we write $\text{extr}_1$ for the channel extraction function used in (3.18).

Explicitly, this means that for $\tau \in \mathcal{M}_1(X \times Y)$ and $z \in X$,

$$\text{Flrn}(\text{extr}_1(\tau)(z)) = \text{extr}_1(\text{Flrn}(\tau))(z).$$

**Proof.** By unfolding the relevant definitions one gets, for $z \in X$,

$$(\text{extr}_1 \circ \text{Flrn})(\tau)(z) = \text{extr}_1 \left( \sum_{x,y} \frac{\tau(x, y)}{t} |_{x,y} \right)(z) \quad \text{where } t = \sum_{x,y} \tau(x, y)$$

$$= \sum_y \left( \sum_{x,y} \frac{\tau(x, y)}{t} |_{y} \right) (y)$$

$$= \sum_y \left( \frac{\tau(z, y)}{\sum_y \tau(z, y)} |_{y} \right) (y)$$

$$= \text{Flrn} \left( \sum_y \frac{\tau(z, y)}{\sum_y \tau(z, y)} |_{y} \right)$$

$$= \text{Flrn}(\text{extr}_1(\tau)(z))$$

$$= (\text{Flrn}^X \circ \text{extr}_1)(\tau)(z).$$

We review what this result means in an earlier illustration.
4.1. Learning a channel from tabular data

Example 4.1.2. We return to the medicine / blood pressure table from Subsection 1.4.1. It involves a joint multiset $\tau \in M([H, L] \times \{1, 2, 3\})$ that captures the table of data:

$$\tau = 10|H, 0\rangle + 35|H, 1\rangle + 25|H, 2\rangle + 5|L, 0\rangle + 10|L, 1\rangle + 15|L, 2\rangle.$$ 

As we have seen in Subsection 1.4.5 the first, multiset extraction captures the two rows of Table 1.4:

$$\text{extr}_1(\tau)(H) = \sum_{x \in M} \tau(H, x)|x\rangle = 10|0\rangle + 35|1\rangle + 25|2\rangle$$
$$\text{extr}_1(\tau)(L) = \sum_{x \in M} \tau(L, x)|x\rangle = 5|0\rangle + 10|1\rangle + 15|2\rangle.$$

We can apply frequentist learning (normalisation) to these two multisets separately, and obtain as probabilistic channel $c : [H, L] \Rightarrow \{1, 2, 3\}$,

$$c(H) = \text{Flrn}(\text{extr}_1(\tau)(H)) = \frac{1}{7}|0\rangle + \frac{1}{7}|1\rangle + \frac{5}{14}|2\rangle$$
$$c(L) = \text{Flrn}(\text{extr}_1(\tau)(L)) = \frac{1}{7}|0\rangle + \frac{1}{7}|1\rangle + \frac{1}{7}|2\rangle.$$

This channel is obtained by following the east-south part of the rectangle in Proposition 4.1.1 above.

If we take the south-east route in the rectangle, we first apply frequentist learning to $\tau$ itself, giving a joint state in $\text{Flrn}(\tau) \in D([H, L] \times \{1, 2, 3\})$. We can then apply disintegration to $\text{Flrn}(\tau)$, using Equation (3.20). Proposition 4.1.1 says that we will get the same channel $c$ in this way.

Thus, the essence of Proposition 4.1.1 is: one can extract a (probabilistic) channel from a table (as joint multiset), either by normalising the rows separately, or by first normalising the entire table and then disintegrating. The outcomes are the same.

Remark 4.1.3. In the south-east route in the diagram in Proposition 4.1.1 we perform disintegration $\text{extr}_1$ after frequentist learning $\text{Flrn}$. Both these operations involve normalisation. These efforts can be combined if we allow ourselves to perform disintegration not only on joint distributions $\omega \in D(X \times Y)$, but also on joint multisets $\tau \in M(X \times Y)$. Concretely, this means that we allow ourselves the use of formulation (3.18) not only for states $\omega$, but also for multisets $\tau$. When we apply this in the context of Example 4.1.2 we directly get
the (same) outcomes that we saw there:

\[ \text{extr}_1(\tau)(H) = \tau[0, 1 \mid 1, 0][H] = \sum_{x \in M} \tau(H, x) \left\{ x \right\} \]

\[ = \frac{10}{10 + 35 + 25} |0\rangle + \frac{35}{10 + 35 + 25} |0\rangle + \frac{25}{10 + 35 + 25} |0\rangle = \frac{1}{6} |0\rangle + \frac{1}{2} |1\rangle + \frac{5}{12} |2\rangle \]

\[ \text{extr}_1(\tau)(L) = \tau[0, 1 \mid 1, 0][L] = \sum_{x \in M} \tau(L, x) \left\{ x \right\} \]

\[ = \frac{5}{5 + 10 + 15} |0\rangle + \frac{10}{5 + 10 + 15} |0\rangle + \frac{15}{5 + 10 + 15} |0\rangle = \frac{1}{6} |0\rangle + \frac{1}{2} |1\rangle + \frac{5}{12} |2\rangle \]

We now also allow ourselves to use the more complicated disintegration expressions \( \tau[\cdots \cdots] \) from Remark 3.5.6 for multisets \( \tau \). This allows us to learn a channel directly from a multidimensional table, as multiset.

**Exercises**

4.1.1 Prove in general what we have claimed and illustrated in Remark 4.1.3, namely that for a multiset \( \tau \in M(X \times Y) \) whose first marginal has full support, one obtains an equality of disintegrations:

\[ \text{extr}_1(\tau) = \text{extr}_1(\text{Flrn}(\tau)) : X \rightarrow Y. \]

**4.2 Learning with missing data**

This section looks at learning from tables in which some entries are missing. We do not treat this topic exhaustively, but instead we elaborate two examples from the literature in order to given an impression of the issues involved.

**4.2.1 Learning with missing data, without prior**

We look at an example from [57, §6.2.1]. It involves pregnancy of cows, which can be deduced from a urine test and a blood test. A simple Bayesian network structure is assumed, which we write as string diagram:

\[ \begin{align*}
\text{Blood Test} & \quad \text{Urine test} \\
B & \quad U \\
P & \quad \text{Pregnancy}
\end{align*} \]

with sets \( P = \{ p, p' \} \)

\( B = \{ b, b' \} \)

\( U = \{ u, u' \} \).
4.2. Learning with missing data

The elements \( p \) and \( p' \) represent ‘pregnancy’ and ‘no pregnancy’, respectively. Similarly, \( b, b^* \) and \( u, u^* \) represent a positive and negative blood/urine test.

The data in [57] involves 5 cases, as given in the following table, where a question mark is used for a missing item.

<table>
<thead>
<tr>
<th>case</th>
<th>Pregnancy</th>
<th>Blood test</th>
<th>Urine test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>?</td>
<td>( b )</td>
<td>( u )</td>
</tr>
<tr>
<td>2</td>
<td>( p )</td>
<td>( b^* )</td>
<td>( u )</td>
</tr>
<tr>
<td>3</td>
<td>( p )</td>
<td>( b )</td>
<td>?</td>
</tr>
<tr>
<td>4</td>
<td>( p )</td>
<td>( b )</td>
<td>( u^* )</td>
</tr>
<tr>
<td>5</td>
<td>?</td>
<td>( b^* )</td>
<td>?</td>
</tr>
</tbody>
</table>

Earlier, we could simply read-off the multiset from a completely filled table, for instance in Subsection 1.4.1. Here, in Table (4.3), this can be done only for the second and fourth line: the second line, for instance, corresponds to a singleton multiset \( 1|p, b^*, u \rangle \). The first step in our analysis is to see this as a tensor of one-dimensional data/multisets, as in:

\[
1|p, b^*, u \rangle = 1|p \rangle \otimes 1|b^* \rangle \otimes 1|u \rangle.
\]

The second step is to use uniform distributions in the \(?\)-case, where an entry is missing. We use these uniform distributions because we assume no prior information about the missing table entries. We can then use the fact that tensoring preserves sums, see Exercise 1.8.15.

For instance, the first line in the above table (4.3) becomes:

\[
(\frac{1}{2}|p \rangle + \frac{1}{2}|p' \rangle) \otimes 1|b \rangle \otimes 1|u \rangle = \frac{1}{2}|p \rangle \otimes 1|b \rangle \otimes 1|u \rangle + \frac{1}{2}|p' \rangle \otimes 1|b \rangle \otimes 1|u \rangle = \frac{1}{2}|p, b, u \rangle + \frac{1}{2}|p', b, u \rangle.
\]

The fifth line has two missing entries so that our analysis involves preservation of convex sums in two tensor coordinates:

\[
\left(\frac{1}{2}|p \rangle + \frac{1}{2}|p' \rangle\right) \otimes \left(\frac{1}{2}|u \rangle + \frac{1}{2}|u' \rangle\right) = \frac{1}{4}|p, b, u \rangle + \frac{1}{4}|p, b, u' \rangle + \frac{1}{4}|p', b, u \rangle + \frac{1}{4}|p', b, u' \rangle.
\]

Adding up these five cases gives a multiset \( \tau \in \mathcal{M}(P \times B \times U) \) of the form:

\[
\tau = \frac{1}{2}|p, b, u \rangle + \frac{1}{2}|p^*, b, u \rangle + 1|p, b^*, u \rangle + \frac{1}{2}|p, b, u \rangle + \frac{1}{2}|p, b, u' \rangle + 1|p, b^*, u' \rangle + \frac{1}{2}|p, b^*, u \rangle + \frac{1}{2}|p^*, b, u \rangle + \frac{1}{2}|p^*, b, u^* \rangle.
\]

We can now learn the two ‘Blood test’ and ‘Urine Test’ channels \( P \rightarrow B \)
and $P \to U$ in (4.2) from this table $\tau$, directly via disintegration — see Exercise 4.1.1 and marginalisation.

- We get $c \coloneqq \tau[0,1,0\,1,0,0] : P \to B$ with:

$$c(p) = \frac{\tau(p,b,u) + \tau(p,b,u^*)}{\tau(p,b,u) + \tau(p,b,u^*) + \tau(p,b^*,u) + \tau(p,b^*,u^*)}$$

$$= \frac{1 + \frac{3}{2} + \frac{1}{4} + \frac{1}{4}}{1 + \frac{3}{2} + \frac{5}{4} + \frac{1}{4}} \cdot b$$

$$= \frac{5}{8} |b^* \rangle + \frac{3}{8} |b^e \rangle.$$

We see that a pregnant cow has a slightly higher probability than a non-pregnant cow of getting a positive blood test.

- In the same way we obtain a channel $d \coloneqq \tau[0,0,1\,1,0,0] : P \to U$ with:

$$d(p) = \frac{9}{16} |u \rangle + \frac{7}{16} |u^* \rangle \quad \text{and} \quad d(p^*) = \frac{7}{16} |u \rangle + \frac{9}{16} |u^* \rangle.$$

Apparantly, the urine test is more likely to be positive for a non-pregnant cow.

These channels $c$ and $d$ coincide with the conditional probabilities computed in [57], without explicitly using tensors of multisets and disintegration.

### 4.2.2 Learning with missing data, with prior

In the previous example we used a uniform distribution for the missing data items — corresponding to the fact that we assumed no prior knowledge. Here we look into an example, taken from [20] §17.3.1, where we do have prior information, namely in the form of a Bayesian network, involving four sets $A = \{a, a^*\}, B = \{b, b^*\}, C = \{c, c^*\}$ and $D = \{d, d^*\}$, in the string diagram:

\[\begin{array}{c}
D \\
\downarrow h \\
B \\
\downarrow f \\
A \\
\downarrow g \\
C \\
\downarrow \omega
\end{array}\]

with

$$\begin{align*}
\omega &= \frac{1}{2} |a \rangle + \frac{1}{2} |a^* \rangle \\
f(a) &= \frac{1}{2} |b \rangle + \frac{1}{2} |b^* \rangle \\
f(a^*) &= \frac{1}{2} |b \rangle + \frac{3}{10} |b^* \rangle \\
h(b) &= \frac{1}{2} |d \rangle + \frac{1}{2} |d^* \rangle \\
h(b^*) &= \frac{1}{2} |d \rangle + \frac{7}{10} |d^* \rangle \\
g(a) &= \frac{1}{2} |c \rangle + \frac{1}{2} |c^* \rangle \\
g(a^*) &= \frac{1}{2} |c \rangle + \frac{7}{10} |c^* \rangle
\end{align*}\]
4.2. Learning with missing data

We first turn this Bayesian network into a joint state \( \tau \in \mathcal{D}(A \times B \times C \times D) \), as interpretation of the accessible string diagram on the left below — obtained from the string diagram in (4.4).

\[
\tau := (\text{id} \otimes \text{id} \otimes \text{swap}) \circ (\text{id} \otimes \text{id} \otimes h \otimes \text{id}) \\
\circ (\text{id} \otimes \Delta_2 \otimes \text{id}) \circ (\text{id} \otimes f \otimes g) \circ \Delta_3) \triangleright \omega
\]

That is:

\[
\tau = \begin{array}{cccc}
A & B & C & D \\
\hline
h & f & g & \omega \\
\end{array}
\]

This distribution \( \tau \) will be used as prior.

The table with data provided in (20) is:

<table>
<thead>
<tr>
<th>case</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>?</td>
<td>b</td>
<td>c</td>
<td>?</td>
</tr>
<tr>
<td>2</td>
<td>?</td>
<td>b</td>
<td>?</td>
<td>d</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>b</td>
<td>c</td>
<td>d</td>
</tr>
<tr>
<td>4</td>
<td>?</td>
<td>b</td>
<td>?</td>
<td>d</td>
</tr>
<tr>
<td>5</td>
<td>?</td>
<td>b</td>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>

At this stage we do not use uniform distributions as fill-in. Instead, we use the Bayesian network to infer probabilities for the missing data. For instance, in the first case we have point predicates \( 1_b \) and \( 1_c \). From the crossover update corollary 2.5.9 we know that this inference can also be done via the joint state \( \tau \), namely as:

\[
\tau_1 := \tau|_{1_b \otimes 1_c \otimes 1_d}
\]

The missing probabilities for \( A, D \) are then obtained via the marginal, as:

\[
\tau_1[1,0,0,1] \\
= 0.1111|a,d| + 0.4444|a,d^+| + 0.08889|a^+,d| + 0.3556|a^+,d^+|
\]

This distribution appears in (20) Fig. 17.4 (a)].
In a similar way we define:

\[ \tau_2 \coloneqq \tau_{1|0|1,0|1}, \quad \tau_3 \coloneqq \tau_{1|1|1,0|1}, \quad \tau_4 \coloneqq \tau_3, \quad \tau_5 \coloneqq \tau_2. \]

These five states \( \tau_1, \ldots, \tau_5 \) are combined in a uniform convex combination, giving a new joint state:

\[ \tau' = \frac{1}{5} \cdot \tau_1 + \frac{1}{5} \cdot \tau_2 + \frac{1}{5} \cdot \tau_3 + \frac{1}{5} \cdot \tau_4 + \frac{1}{5} \cdot \tau_5 = \frac{1}{5} \cdot \tau_1 + \frac{2}{5} \cdot \tau_2 + \frac{2}{5} \cdot \tau_3. \]

From \( \tau' \) we can obtain a newly learned interpretation of the Bayesian network (4.4), via marginalisation and disintegration. In detail, the new distribution \( \omega' \) on \( A \) is:

\[ \omega' \coloneqq \tau'[1,0,0,0] = 0.4208|a\rangle + 0.5792|a^\perp\rangle. \]

With newly learned channels:

\[ f' \coloneqq \tau'[0,1,0,0|1,0,0,0] \begin{cases} f'(a) = 0.8841|b\rangle + 0.1159|b^\perp\rangle \\ f'(a^\perp) = 0.3937|b\rangle + 0.6063|b^\perp\rangle \end{cases} \]

\[ g' \coloneqq \tau'[0,0,1|1,0,0,0] \begin{cases} g'(a) = 0.4259|c\rangle + 0.5741|c^\perp\rangle \\ g'(a^\perp) = 0.6664|c\rangle + 0.3336|c^\perp\rangle \end{cases} \]

\[ h' \coloneqq \tau'[0,0,0,1|0,1,0,0] \begin{cases} h'(b) = 0.06667|d\rangle + 0.9333|d^\perp\rangle \\ h'(b^\perp) = 1|d\rangle. \end{cases} \]

These outcomes are as in [20, Fig. 17.5] — up to some small differences, probably due to rounding.

Interestingly, Table (4.5) gives no information about \( A \), but still we learn a new distribution \( \omega' \) on \( A \), basically from the prior information \( \tau \).

**Exercises**

4.2.1 Show that the pregnancy distribution in Subsection 4.2.1, obtained from Table (4.3), is \( \frac{4}{9}|p\rangle + \frac{1}{9}|p^\perp\rangle \).

4.2.2 Calculate in detail that, in Subsection 4.2.2,

\[ \tau_1[1,0,0,1] = \frac{1}{9}|a,d\rangle + \frac{4}{9}|a,d^\perp\rangle + \frac{4}{45}|a^\perp,d\rangle + \frac{16}{45}|a^\perp,d^\perp\rangle. \]

4.2.3 Apply the approach of Subsection 4.2.2 to the example in Subsection 4.2.1, starting from the Bayesian network 4.2 interpreted with only uniform distributions. Check that it gives the same outcome as given in Subsection 4.2.1.
4.3 Increasing validity via updating

In the beginning of this section we have described a basic form of learning, where the aim is to increase a validity $\omega \models p$ into a higher validity $\omega' \models p$ by changing $\omega$ into a ‘better’ state $\omega'$. The main result in this section is that one can take $\omega' = \omega|_p$, the conditioning of $\omega$ by $p$. This requires that $p$ is a factor (has non-negative values), see Section 2.3.

We provide a proof that $\omega|_p$ increases the validity of $p$, see Theorem 4.3.1 below. Another proof will be given later on, in Example 4.4.3 via a more general ‘sum-increase’ methodology, that will appear in Section 4.4.

The main result of this section, Theorem 4.3.1, can be used quite generally, to derive various other forms of validity increases, not only by updating states, but also by updating channels.

**Theorem 4.3.1.** Let $\omega \in \mathcal{D}(X)$ and $p \in \text{Obs}(X)$ be a state and an observable on the same set $X$. Then there are inequalities between validities:

1. $(\omega \models p)^2 = \omega \otimes \omega \models p \otimes p \leq \omega \models p \& p$;
2. $\omega \models p \leq \omega|_p \models p$, when $p$ is a factor.

The second point says that we can increase the validity of $p$ in $\omega$ by updating $\omega$ with $p$. This confirms the intuition that $p$ becomes ‘more true’ by moving from state $\omega$ to $\omega|_p$.

An obvious generalisation of the first point to different predicates does not hold, see Exercise 4.3.1 below.

**Proof.** 1 We first give an elementary proof, and then provide an alternative proof using variance. For the first proof we rely on the square of sums formula:

$$(v_1 + \cdots + v_n)^2 = \sum_i v_i^2 + \sum_{j \neq i} 2v_i v_j. \quad (4.6)$$

Let $\text{supp}(\omega) = \{x_1, \ldots, x_n\}$ and write $\omega_i = \omega(x_i)$ and $p_i = p(x_i)$. We then need to prove the inequality in the middle:

$$(\omega \models p)^2 = (\sum_i \omega_i p_i)^2 \leq \sum_i \omega_i p_i^2 = \omega \models p \& p.$$
We proceed as follows.

\[
\sum_i \omega_i p_i^2 = \sum_i \omega_i \omega_i p_i^2 + (1 - \omega_i)\omega_i p_i^2 \\
= \sum_i \omega_i^2 p_i^2 + \sum_i (\sum_j \omega_j)\omega_i p_i^2 \\
\geq \sum_i \omega_i^2 p_i^2 + \sum_i \sum_j \omega_i \omega_j (p_i^2 + p_j^2) \quad \text{by re-ordering} \\
\geq \sum_i (\omega_i p_i)^2 + \sum_{i,j} 2(\omega_i p_i)(\omega_j p_j) \\
= (\sum_i \omega_i p_i)^2 \quad \text{by (4.6).}
\]

The marked inequality (≥) holds because:

\[
0 \leq (p_i - p_j)^2 = p_i^2 - 2p_i p_j + p_j^2 \quad \text{so} \quad 2p_i p_j \leq p_i^2 + p_j^2.
\]

There is an alternative way to prove this first point. We know from Definition 2.9.1 that the variance \( \text{Var}(\omega, p) \) of the random variable \((\omega, p)\) is non-negative, since it is defined as the validity of a factor. But Lemma 2.9.3 tells that \( \text{Var}(\omega, p) = (\omega \models p^2) - (\omega \models p)^2 \), so that \((\omega \models p^2) \geq (\omega \models p)^2\), as claimed above.

2 We now assume that \(p\) is a factor and that the validity \(\omega \models p\) is non-zero, so that \(\omega | p\) is defined. The first equation below is Bayes’ product rule, see Proposition 2.3.3 (2); the subsequent inequality comes from the previous point.

\[
\omega_p | p = \frac{\omega | p & p \models p}{\omega \models p} \geq \frac{(\omega \models p)^2}{\omega \models p} = \omega \models p.
\]

By \(n\)-fold iterating the updating \(\omega_p \cdots | p = \omega_p \cdots & p = \omega_p\) we reach a state where the factor \(p\) has its maximum validity.

The previous theorem gives important leads about how to increase validities by updating both states and channels (see Definition 2.3.1).

**Corollary 4.3.2.** Let \(c: X \rightarrow Y\) be a channel with \(\omega \in \mathcal{D}(X)\) and \(q \in \text{Fact}(Y)\).

1 There are inequalities between validities:

\[
\begin{align*}
\vdash c \gg \omega \models q \\
\vdash c \gg (\omega|_{c \models q}) \models q \\
\vdash c|_q \gg \omega \models q \\
\vdash c|_q \gg (\omega|_{c \models q}) \models q
\end{align*}
\]
4.3. Increasing validity via updating

2 For an additional channel \( d: X \rightarrow Z \) and a factor \( r \in \text{Fact}(Z) \),

\[
\langle c, d \rangle \gg \omega \models q \otimes r \\
\langle c \lceil q, d \rceil \rangle \gg (\omega \mid_{c \ll q \& d \ll r}) \models q \otimes r \\
\langle c \rceil_{q}, d \lceil r \rangle \gg (\omega \mid_{c \ll q \& d \ll r}) \models q \otimes r
\]

**Proof.** For both points it suffices to prove the upper two inequalities.

1 We do the upper-left inequality first:

\[
c \gg \omega \models q = \omega \models c \ll q \quad \text{by Proposition 2.4.3} \\
\leq \omega \mid_{c \ll q} \models c \ll q \quad \text{by Theorem 4.3.1 (2)} \\
= c \gg (\omega \mid_{c \ll q}) \models q \quad \text{by Proposition 2.4.3 again.}
\]

For proving the upper-right inequality we unpack validity:

\[
c \lceil q \gg \omega \models q = \sum_{y} (c \lceil q \gg \omega)(y) \cdot q(y) \\
= \sum_{x,y} \omega(x) \cdot c_{q}(x)(y) \cdot q(y) \\
= \sum_{x,y} \omega(x) \cdot \frac{c(x)(y) \cdot q(y)}{c(x) \models q} \cdot q(y) \\
= \sum_{x} \omega(x) \cdot \frac{c(x) \models q}{c(x) \models q} \cdot q(y) \\
\geq \sum_{x} \omega(x) \cdot \frac{(c(x) \models q)^{2}}{c(x) \models q} \\
= \sum_{x} \omega(x) \cdot (c(x) \models q) \\
= \sum_{x} \omega(x) \cdot c(x)(y) \cdot q(y) \\
= c \gg \omega \models q.
\]

2 The upper-left inequality follows from the upper-left inequality in the previous point since \( \langle c, d \rangle \ll (q \otimes r) = (c \ll q) \& (d \ll r) \). This follows from Exercise 2.4.3:

\[
\langle c, d \rangle \ll (q \otimes r) = ((c \otimes d) \ast \Delta) \ll (q \otimes r) \\
= \Delta \ll ((c \otimes d) \ll (q \otimes r)) \\
= \Delta \ll ((c \ll q) \otimes (d \ll r)) \\
= (c \ll q) \& (d \ll r).
\]
Similarly, one obtains the upper-right inequality from point (1) and the equation \( \langle c, d \rangle_{q \otimes r} = \langle c | q, d | r \rangle \), which holds by Exercise 2.3.9.

Recall that observables carry a partial order \( \leq \), which is obtained pointwise, from the order of the reals: \( p \leq q \) iff \( p(x) \leq q(x) \) for all \( x \). The above results about inequalities between validities also have consequences for predicates.

**Corollary 4.3.3.** For a channel \( c : X \to Y \) and a factor \( q \) on \( Y \) one has:

\[
c \ll q \leq c q \ll q.
\]

**Proof.** For each \( x \in X \),

\[
(c \ll q)(x) = 1|x \models c \ll q
= c \gg (1|x)) \models q
\leq c q \gg (1|x)) \models q \quad \text{by Corollary 4.3.2 (1)}
= 1|x \models c q \ll q
= (c q \ll q)(x).
\]

**Exercises**

4.3.1 Check that Theorem 4.3.1 (1) does not generalise to:

\[(\omega \models p) \cdot (\omega \models q) \leq (\omega \models p \& q).\]

Take for instance \( \omega = \frac{1}2|H\rangle + \frac{1}2|T\rangle \) with \( p(H) = \frac{1}2, p(T) = \frac{4}5 \) and \( q(H) = \frac{1}2, q(T) = \frac{1}2.\)

### 4.4 A general sum-increase method

The previous section has introduced several inequalities between validities. They were all derived from a basic observation, in Theorem 4.3.1 (1). This section introduces a more general approach to derive inequalities, from [7]. It forms the basis for many learning methods. This approach involves a generic sum-increase result, see Lemma 4.4.2 below. In many situations a concrete description of this increase can be given via some basic real analysis. This will be illustrated, among others, by re-deriving, in a completely different manner, some of the inequalities of the previous section.

We begin with a classical result in this setting that we need below.

**Lemma 4.4.1** (Jensen’s inequality). Let \( \omega \in D(X) \) be state with a factor
4.4. A general sum-increase method

$p: X \to \mathbb{R}_{\geq 0}$ on its domain which is positive (non-zero) on the support of $\omega$. For each function $f: \mathbb{R}_{\geq 0} \to \mathbb{R}$ with $f'' < 0$ one has:

$$f(\omega \models p) \geq \omega \models (f \circ p).$$

The inequality is strict, except in trivial cases.

This result holds in particular for $f = \ln$, the natural logarithm.

The proof is standard but is included nevertheless, for convenience of the reader.

**Proof.** It suffices to prove that for $a_1, \ldots, a_n \in \mathbb{R}_{\geq 0}$ and $r_1, \ldots, r_n \in [0, 1]$ with $\sum_ir_i = 1$ one has $f(\sum_ir_ia_i) \geq \sum_ir_if(a_i)$. We shall provide a proof for $n = 2$. The inequality is easily extended to $n > 2$, by induction.

So let $a, b \in \mathbb{R}_{\geq 0}$ be given, with $r \in [0, 1]$. We need to prove $f(ra + (1-r)b) \geq rf(a) + (1-r)f(b)$. The result is trivial if $a = b$ or $r = 0$ or $r = 1$. So let, without loss of generality, $a < b$ and $r \in (0, 1)$. Write $c = ra + (1-r)b = b - r(b-a)$, so that $a < c < b$. By the mean value theorem we can find $a < u < c$ and $c < v < b$ with:

$$\frac{f(c) - f(a)}{c-a} = f'(u) \quad \text{and} \quad \frac{f(b) - f(c)}{b-c} = f'(v)$$

Since $f'' < 0$ we have that $f'$ is strictly decreasing, so $f'(u) > f'(v)$ because $u > v$. We can write:

$$c - a = (r - 1)a + (1-r)b = (1-r)(b-a) \quad \text{and} \quad b - c = r(b-a).$$

From $f'(u) > f'(v)$ we deduce inequalities:

$$\frac{f(c) - f(a)}{(1-r)(b-a)} > \frac{f(b) - f(c)}{r(b-a)} \quad \text{i.e.} \quad rf(c) - f(a) > (1-r)f(b) - f(c).$$

By reorganising the latter inequality we get $f(c) > rf(a) + (1-r)f(b)$, as required. 

We now come to the main technical result which we shall call the sum-increase lemma. It is a special (discrete) case of a more general result [7, Thm. 2.1]. In the previous section we concentrated on increasing validities $\omega \models p = \sum_y \omega(y) \cdot p(y)$. The next result describes how to find increases for sum expressions in general.

**Lemma 4.4.2.** Let $X, Y$ be finite sets, and let $F: X \times Y \to \mathbb{R}_{\geq 0}$ be a given function. For each $x \in X$, write $F_1(x) := \sum_y F(x,y)$. Assume that there is an $x' \in X$ with:

$$x' \in \arg\max_z G(x, z) \quad \text{where} \quad G(x, z) := \sum_{y \in Y} F(x, y) \cdot \ln(F(z, y)).$$
Then $F_1(x') \geq F_1(x)$. We often say informally that $x'$ is better $x$.

In general, the inequality $\geq$ in this lemma gives a strict increase, like in Jensen’s inequality in Lemma 4.4.1. It forms the basis for many forms of learning. An actual maximum $x'$ can in many situations be determined analytically — using the Lagrange multiplier method — but it need not be unique and it can be a local maximum. This will be illustrated in the remainder of this section. We will see that several validity increases from the previous section can be obtained via the above lemma.

**Proof.** Let $x'$ be the element where $G(x, -) : Y \to \mathbb{R}_{\geq 0}$ takes its maximum. This $x'$ satisfies $F_1(x') \geq F_1(x)$, since:

$$\ln \left( \frac{F_1(x')}{F_1(x)} \right) = \ln \left( \sum_{y} \frac{F(x', y)}{F_1(x)} \right)$$

$$= \ln \left( \sum_{y} \frac{F(x, y)}{F_1(x)} \frac{F(x', y)}{F(x, y)} \right)$$

$$\geq \sum_{y} \frac{F(x, y)}{F_1(x)} \cdot \ln \left( \frac{F(x', y)}{F(x, y)} \right)$$

by Lemma 4.4.1

$$= \frac{1}{F_1(x)} \sum_{y} F(x, y) \cdot \left( \ln (F(x', y)) - \ln (F(x, y)) \right)$$

$$= \frac{1}{F_1(x)} \left( G(x, x') - G(x, x) \right) \geq 0. \quad \Box$$

**Example 4.4.3.** Suppose we have a validity $\omega \models p$, where $\omega \in \mathcal{D}(Y)$ and $p \in \text{Fact}(Y)$, which we would like to increase, by changing the state $\omega$, while keeping $p$ fixed. We apply Lemma 4.4.2 with $F : \mathcal{D}(Y) \times Y \to \mathbb{R}_{\geq 0}$ given by $F(\omega, y) = \omega(y) \cdot p(y)$. Then $F_1(\omega) = \sum_y F(\omega, y) = \sum_y \omega(y) \cdot p(y) = \omega \models p$.

We have $G(\omega, \omega') = \sum_y \omega(y) \cdot p(y) \cdot \ln(\omega'(y) \cdot p(y))$ and we wish to find a maximum of $G(\omega, -) : \mathcal{D}(Y) \to \mathbb{R}_{\geq 0}$. Let $Y = \{y_1, \ldots, y_n\}$; we will use variables $v_i$ for the values $\omega'(y_i)$ that we wish to find, by seeing $G(\omega, -)$ as a function $\mathbb{R}^n \to \mathbb{R}_{\geq 0}$ with constraints.

These constraints are handled via the Lagrange multiplier method for finding the maximum (see e.g. §2.2). We keep $\omega$ fixed and consider a new function $H$, also known as the Lagrangian, with an additional parameter $\kappa$.

$$H(\vec{v}, \kappa) := G(\omega, \vec{v}) - \kappa \cdot (\ell) \cdot (\sum_i v_i - 1)$$

$$= \sum_i \omega(y_i) \cdot p(y_i) \cdot \ln (v_i \cdot p(y_i)) - \kappa \cdot (\sum_i v_i - 1). \quad \Box$$
4.4. A general sum-increase method

The partial derivatives of $H$ are:

\[
\frac{\partial H}{\partial v_i}(\vec{v}, \kappa) = \frac{\omega(y_i) \cdot p(y_i)}{v_i \cdot p(y_i)} \cdot (\omega(y_i) \cdot p(y_i)) = \frac{\omega(y_i) \cdot p(y_i)}{v_i} - \kappa
\]

\[
\frac{\partial H}{\partial \kappa}(\vec{v}, \kappa) = (\sum_i v_i) - 1.
\]

Setting all these derivatives to zero yields:

\[
1 = \sum_i v_i = \sum_i \frac{\omega(y_i) \cdot p(y_i)}{\kappa} = \frac{\omega \upharpoonright p}{\kappa}.
\]

Hence $\kappa = \omega \upharpoonright p$ and thus:

\[
v_i = \frac{\omega(y_i) \cdot p(y_i)}{\kappa} = \frac{\omega(y_i) \cdot p(y_i)}{\omega \upharpoonright p} = \omega \upharpoonright p(y_i).
\]

Hence, Lemma 4.4.2 gives as 'better' state the updated version $\omega \upharpoonright p$ of $\omega$ with $p$, as in Theorem 4.3.1 [1], where we already saw the inequality $\omega \upharpoonright p \upharpoonright p \geq \omega \upharpoonright p$.

**Example 4.4.4.** We will now try to find a ‘better’ version of a channel $c$ in a situation $c \gg \omega \upharpoonright q$, where $c : X \rightarrow Y$, $\omega \in \mathcal{D}(X)$ and $q \in \text{Fact}(Y)$. We define $F : \mathcal{D}(Y)^X \times X \times Y \rightarrow \mathbb{R}_{\geq 0}$ as:

\[
F(c, x, y) := \omega(x) \cdot c(x)(y) \cdot q(y)
\]

Then $F_1(c) = \sum_{x,y} F(c, x, y) = \sum_c (c \gg \omega)(y) \cdot q(y) = c \gg \omega \upharpoonright q$.

Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$. We shall use variables $z_{ij}$ for $c'(x_i)(y_j)$. We can thus describe $G(c, -) : \mathbb{R}^{\text{comb}} \rightarrow \mathbb{R}_{\geq 0}$ as:

\[
G(c, \vec{z}) = \sum_{i,j} F(c, x_i, y_j) \cdot \ln(F(c, x_i, y_j))
\]

\[
= \sum_{i,j} \omega(x_i) \cdot c(x_i)(y_j) \cdot q(y_j) \cdot \ln(\omega(x_i) \cdot z_{ij} \cdot q(y_j)).
\]

We now use the Lagrange multiplier method with $n$ additional variables $\vec{\kappa}$ in a new function:

\[
H(\vec{z}, \vec{\kappa}) := \sum_{i,j} \omega(x_i) \cdot c(x_i)(y_j) \cdot q(y_j) \cdot \ln(\omega(x_i) \cdot z_{ij} \cdot q(y_j))
\]

\[
- \sum_i \kappa_i \cdot ((\sum_j z_{ij}) - 1).
\]

Its partial derivatives are:

\[
\frac{\partial H}{\partial z_{ij}}(\vec{z}, \vec{\kappa}) = \frac{\omega(x_i) \cdot c(x_i)(y_j) \cdot q(y_j)}{\omega(x_i) \cdot z_{ij} \cdot q(y_j)} - \kappa_i
\]

\[
= \frac{\omega(x_i) \cdot c(x_i)(y_j) \cdot q(y_j)}{z_{ij}} - \kappa_i
\]

\[
\frac{\partial H}{\partial \kappa_j}(\vec{z}, \vec{\kappa}) = (\sum_j z_{ij}) - 1.
\]
Setting all these derivatives to zero yields for each $i$,

$$1 = \sum_j z_{ij} = \sum_j \frac{\omega(x_i) \cdot c(x_i)(y_j) \cdot q(y_j)}{\kappa_i} = \frac{\omega(x_i) \cdot (c(x_i) \gg q)}{\kappa_i}.$$ 

Hence $\kappa_i = \omega(x_i) \cdot (c(x_i) \gg q)$ so that:

$$z_{ij} = \frac{\omega(x_i) \cdot c(x_i)(y_j) \cdot q(y_j)}{\kappa_i} = \frac{\omega(x_i) \cdot c(x_i)(y_j) \cdot q(y_j)}{\omega(x_i) \cdot (c(x_i) \gg q)} = \frac{c(x_i)(y_j) \cdot q(y_j)}{c(x_i) \gg q} = c_i (y_j) \cdot q(y_j) = c_i (y_j).$$

Thus, the ‘better’ channel than $c$ in the situation $c \gg \omega \gg q$ is $c_i$. This matches Corollary 4.3.2 (1) where we already saw that $c_i \gg \omega \gg q \gg c \gg \omega \gg q$. 

In the description of Lemma 4.4.2 the set $X$ may be a product of other sets, so that the sum-increase approach also works for multiple channels and/or states — giving a better version for each of them. This will be illustrated next.

Example 4.4.5. We again consider a situation $c \gg \omega \gg q$ as in Example 4.4.3 but now we wish to get better versions of both $c$ and $\omega$. The situation is much as before, except that we now use two channel arguments: we define $F : D(Y)^X \times D(X) \times X \times Y \to \mathbb{R}_{\geq 0}$ as:

$$F(c, \omega, x, y) := \omega(x) \cdot c(x)(y) \cdot q(y).$$

Then $F(c, \omega) = \sum_{x,y} F(c, \omega, x, y) = \sum_{x,y} (c \gg \omega)(y) \cdot q(y) = c \gg \omega \gg q.$

Let $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$. We shall use variables $z_{ij}$ for $c'(x_i)(y_j)$ and $w_i$ for $\omega'(x_i)$. We can thus describe $G(c, \omega, -,-) : \mathbb{R}^{m \times n} \times \mathbb{R}^{n} \to \mathbb{R}_{\geq 0}$ as:

$$G(c, \omega, \vec{z}, \vec{w}) = \sum_{i,j} F(c, \omega, x_i, y_j) \cdot \ln (F(z_i \vec{w}, x_i, y_j)) = \sum_{i,j} \omega(x_i) \cdot c(x_i)(y_j) \cdot q(y_j) \cdot \ln (w_i \cdot z_{ij} \cdot q(y_j)).$$

We again use the Lagrange multiplier method, now with variables $\vec{\kappa}, \lambda$ in the function:

$$H(\vec{z}, \vec{w}, \vec{\kappa}, \lambda) := \sum_{i,j} \omega(x_i) \cdot c(x_i)(y_j) \cdot q(y_j) \cdot \ln (w_i \cdot z_{ij} \cdot q(y_j)) - \sum_i \kappa_i \cdot ((\sum_j z_{ij}) - 1) - \lambda \cdot ((\sum_i w_i) - 1).$$
4.4. A general sum-increase method

The partial derivatives are:

\[
\frac{\partial H}{\partial z_{ij}}(\vec{z}, \vec{w}, \vec{\kappa}, \lambda) = \frac{\omega(x_i) \cdot c(x_i)(y_j) \cdot q(y_j)}{w_i \cdot q(y_j)} - \kappa_i
\]

\[
\frac{\partial H}{\partial w_i}(\vec{z}, \vec{w}, \vec{\kappa}, \lambda) = \sum_j \omega(x_i) \cdot c(x_i)(y_j) \cdot q(y_j) - \kappa_i
\]

\[
\frac{\partial H}{\partial \kappa_i}(\vec{z}, \vec{w}, \vec{\kappa}, \lambda) = (\sum_j z_{ij}) - 1
\]

\[
\frac{\partial H}{\partial \lambda}(\vec{z}, \vec{w}, \vec{\kappa}, \lambda) = (\sum_i w_i) - 1.
\]

By taking all these derivatives to be zero we can derive

\[
z_{ij} = c_{ij}(x_i)(y_j),\] as

in Example 4.4.4. We claim that we can also derive

\[
w_i = \omega_i \cdot c_{i} \ll q(x_i),\]

in the following way.

\[
1 = \sum_i w_i = \sum_i \frac{\omega(x_i) \cdot (c \ll q)(x_i)}{\lambda} = \frac{\omega \gg c \ll q}{\lambda}.
\]

Thus \( \lambda = \omega \gg c \ll q \) and so:

\[
w_i = \frac{\omega(x_i) \cdot (c \ll q)(x_i)}{\lambda} = \frac{\omega(x_i) \cdot (c \ll q)(x_i)}{\omega \gg c \ll q} = \omega_i \cdot c_{i} \ll q(x_i).
\]

Thus, we rediscover the inequality \( c_{i} \gg \omega \gg c \ll q \gg \omega \gg q \) from Corollary 4.3.2[1], in which better versions of both the channel \( c \) and the state \( \omega \) are used.

The previous illustrations involved inequalities that we already knew. We conclude this section with a new result.

**Proposition 4.4.6.** Let \( \omega \in \mathcal{D}(X) \) be state on a finite set \( X \) and let \( p_1, \ldots, p_n \) be factors on \( X \), all with non-zero validity \( \omega \gg p_i \).

1 The state \( \omega' = \sum \frac{1}{n} \cdot \omega \mid p_i \), satisfies:

\[
\prod_i (\omega' \gg p_i) \geq \prod_i (\omega \gg p_i).
\]

2 For \( k_1, \ldots, k_n \in \mathbb{N} \) write \( k = \sum k_i \) and \( \omega' = \sum \frac{1}{k} \cdot \omega \mid p_i \), where we assume \( k > 0 \). Then:

\[
\prod_i (\omega' \gg p_i)^{k_i} \geq \prod_i (\omega \gg p_i)^{k_i}.
\]
4.4.2 Prove Proposition 4.4.6 (1) in general, not just for \( n \).

**Proof.** We shall do the proof for \( n = 2 \) and leave the general case as exercise to the reader. We use Lemma 4.4.2 with function \( F : \mathcal{D}(X) \times X \times X \to \mathbb{R}_{\geq 0} \) given by:

\[
F(\omega, x, y) := \omega(x) \cdot p_1(x) \cdot \omega(y) \cdot p_2(y).
\]

Then by distributivity of multiplication over addition:

\[
\sum_{x, y} F(\omega, x, y) = (\sum_x \omega(x) \cdot p_1(x)) \cdot (\sum_y \omega(y) \cdot p_2(y))
= (\omega \models p_1) \cdot (\omega \models p_2).
\]

Let \( X = \{x_1, \ldots, x_n\} \) and let the function \( H \) be given by:

\[
H(\vec{w}, \lambda) := \sum_{i, j} F(\omega, x_i, x_j) \cdot \ln (w_i \cdot p_1(x_i) \cdot w_j \cdot p_2(x_j)) - \lambda \cdot ((\sum_i w_i) - 1).
\]

Then:

\[
\frac{\partial H}{\partial w_k}(\vec{w}, \lambda) = \sum_i \frac{F(\omega, x_i, x_k) + F(\omega, x_i, x_k)}{w_k} - \lambda
\]

Setting these to zero gives:

\[
1 = \sum_k w_k = \frac{\sum_{i, k} F(\omega, x_i, x_k) + F(\omega, x_i, x_k)}{\lambda} = \frac{2 \cdot (\omega \models p_1) \cdot (\omega \models p_2)}{\lambda}.
\]

Hence \( \lambda = 2 \cdot (\omega \models p_1) \cdot (\omega \models p_2) \) so that:

\[
w_k = \frac{\sum_{i, k} F(\omega, x_i, x_k) + F(\omega, x_i, x_k)}{\lambda} = \frac{1}{2} \cdot \omega(x_k) \cdot p_1(x_k) \cdot (\omega \models p_2) + \frac{1}{2} \cdot (\omega \models p_1) \cdot \omega(x_k) \cdot p_2(x_k)
= \frac{1}{2} \cdot \omega(x_k) \cdot p_1(x_k) + \frac{1}{2} \cdot \omega(x_k) \cdot p_2(x_k)
= \frac{1}{2} \cdot \omega(x_k) + \frac{1}{2} \cdot \omega(x_k).
\]

2. Directly from (1), by writing \( \prod_i (\omega' \models p_i)^k \) as a \( k \)-fold product, for \( k = \sum_i k_i \).

**Exercises**

4.4.1 Check that the better channel that appears in Example 4.4.5 is \( c_{\lambda^q} \), like in Example 4.4.4

4.4.2 Prove Proposition 4.4.6 (1) in general, not just for \( n = 2 \).
4.5 Learning Markov chains

The current and next section continue the study of Markov chains and hidden Markov models (HMMs) from Section 3.4. This section will introduce learning for Markov chains, whereas the next section extends the approach to hidden Markov models.

We recall that a Markov chain is a pair \((\sigma, t)\) where \(\sigma \in \mathcal{D}(X)\) is the initial state and \(t: X \rightarrow X\) is the transition channel. It can be seen as a special case of an HMM, where the emission channel is the identity. Thus we will use validity for HMMs also for Markov chains, where \((\sigma, t, \text{id}) \models \vec{p}\) should be understood as \((\sigma, t, \text{id}) \models \vec{p}\), in the sense of Definition 3.4.2.

Below we first look at the simple situation where we like to learn a Markov chain directly from (point) observations, in the style of frequentist learning Flrn. Subsequently we look at the more difficult situation of increasing the validity of a given sequence of observations. This will be done via the sum-increase approach of the previous section. It leads to a channel-based description of the Baum-Welch algorithm [7] (for Markov chains).

4.5.1 Learning by counting elements, without priors

We consider an example. Let \(X = \{A, C, G, T\}\) the letters for the four DNA nucleobase elements adenine (A), cytosine (C), guanine (G) and thymine (T). Suppose we are given two 8-letter sequences \(\alpha_1, \alpha_2 \in \mathcal{L}(X)\), namely:

\[
\]

What can we learn from this? In particular, is there a Markov chain that (best) fits these data?

For the initial state we only have the first entries in the two sequences, namely \(C\) and \(A\). This gives a multiset \(1|A\rangle + 1|C\rangle\), which yields the initial state \(\sigma = \frac{1}{2}|A\rangle + \frac{1}{2}|C\rangle\), via frequentist learning Flrn.

For the transition channel we turn the above two sequences of letters into sequences of consecutive pairs of letters, as in:

\[
[(C, G), (G, A), (A, A), (A, T), (T, T), (T, G), (G, C)] \\
[(A, C), (C, T), (T, C), (C, C), (C, A), (A, T), (T, G)]
\]

We combine these two lists and turn them into a multiset \(\psi \in \mathcal{M}(X \times X)\) of pairs:

\[
\psi = 1|A, A\rangle + 1|A, C\rangle + 2|A, T\rangle + 1|C, A\rangle + 1|C, C\rangle + 1|C, G\rangle + 1|C, T\rangle \\
+ 1|G, A\rangle + 1|G, C\rangle + 1|T, C\rangle + 2|T, G\rangle + 1|T, T\rangle.
\]
Applying frequentist learning to \( \psi \) gives a joint state \( Flm(\psi) \in \mathcal{D}(X \times X) \), from which we can extract the transition channel \( t : X \rightarrow X \) by disintegration. By Remark 4.1.3, we also extract directly from \( \psi \). This yields:

\[
\begin{align*}
t(A) &= \frac{1}{2}|A| + \frac{1}{4}|C| + \frac{1}{4}|T| \\
t(C) &= \frac{1}{2}|A| + \frac{1}{4}|C| + \frac{1}{4}|G| + \frac{1}{4}|T| \\
t(G) &= \frac{1}{2}|A| + \frac{1}{4}|C| + \frac{1}{4}|G| + \frac{1}{4}|T|.
\end{align*}
\]

We then obtain validities:

\[
\begin{align*}
(\sigma, t) \models \alpha_1 &= \frac{1}{2} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} = \frac{19683}{2^{18}} \approx 0.000488 \\
(\sigma, t) \models \alpha_2 &= \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} = \frac{1}{8192} \approx 0.000122.
\end{align*}
\]

Of course, one can also calculate the validities of other sequences in the Markov chain \((\sigma, t)\) learned from \( \alpha_1, \alpha_2 \).

### 4.5.2 Finding a better Markov chain

We are going to apply the techniques from Section 4.4 to learn a new, better, Markov chain, starting from a given (prior) one. This given chain will be written as \((\sigma, t)\), with initial state \( \sigma \in \mathcal{D}(X) \) and transition channel \( t : X \rightarrow X \). We further assume that a list of \( \ell \) factors \( \beta = p_1, \ldots, p_\ell \in \text{Fact}(X) \) is given, of consecutive observations, from which we like to learn. Our aim is to find a better Markov chain \((\sigma', t')\) with:

\[
(\sigma', t') \models \beta \geq (\sigma, t) \models \beta.
\]

We recall from Definition 4.3.4 the filtered sequence of states \( \sigma_1, \ldots, \sigma_{t+1} \in \mathcal{D}(X) \) with \( \sigma_1 := \sigma \) and \( \sigma_{t+1} := t \gg (\sigma|_{t'}) \). We introduce two new sequences, of factors \( q_1, \ldots, q_{t+1} \in \text{Fact}(X) \), and of states \( \gamma_1, \ldots, \gamma_{t+1} \in \mathcal{D}(X) \). First, the factors are defined, ‘backwards’, as in:

\[
q_{t+1} = 1 \quad \text{and} \quad q_{t-1} := p_{t-1} \& (t \ll q_t). \quad (4.7)
\]

We can now define the states \( \gamma_i \), for \( 1 \leq i \leq \ell \), via entry-wise updating, simply as:

\[
\gamma_i := \sigma_i|_{q_i} \quad \text{and} \quad \gamma := \sum_{1 \leq i \leq \ell} \gamma_i. \quad (4.8)
\]

Each \( \gamma_i \) is a state, by construction, but \( \gamma \), as a sum of states, is a multiset.

**Theorem 4.5.1.** In the above setting, starting from a Markov chain \((\sigma, t)\) we define the Baum-Welch improvement of \((\sigma, t)\) with factors \( \beta = p_1, \ldots, p_\ell \) as:

\[
\text{BW}(\sigma, t, \beta) := (\sigma', t'), \quad \text{where:} \quad \sigma' := \gamma_1 \quad t'(x) := \sum_{1 \leq i \leq \ell} \frac{\gamma_i(x)}{\gamma(x)} \cdot t_{q_i}(x).
\]

Then: \( \text{BW}(\sigma, t, \beta) \models \beta \geq (\sigma, t) \models \beta \).
This result describes just one improvement step. What is called the Baum-Welch algorithm for Markov chains involves repeating this step multiple times, with the same factors \( \bar{p} \). The number of repetitions may depend on some precision requirement, like the increase in validity being smaller than some epsilon value.

The rest of this section will be devoted mostly to the proof of this result, using Lemma \[4.4.2\].

We start with several technical properties for the \( q \)’s and \( y \)’s in definitions \[4.7\] and \[4.8\]. They are collected below, for later use.

**Lemma 4.5.2.** In the above situation,

1. The (prior) validity \((\sigma, t) \models \bar{p}\) equals \( \sigma_1 \models q_i \);
2. \( \sigma_{i+1} \models q_{i+1} = \frac{\sigma_j \models q_i}{\sigma_j \models p_i} \) and thus \( \sigma_{i+1} \models q_i = \frac{\sigma_j \models q_i}{\prod_{j \in i} \sigma_j \models p_j} \);
3. \( y_i(x) = \frac{(\prod_{j \in i} \sigma_j \models p_j) \cdot \sigma_i(x) \cdot p_i(x) \cdot (t \ll q_{i+1}(x))}{\sigma_1 \models q_1} \).

**Proof.** The second part of point 2 follows by induction.

We have:

\[
\gamma_i(x) = \frac{\sigma_i(x) \cdot q_i(x)}{\sigma_j \models q_i} = \frac{\sigma_i(x) \cdot p_i(x) \cdot (t \ll q_{i+1}(x))}{\sigma_i(x) \cdot \prod_{j \in i} \sigma_j \models p_j} \quad \text{by 2}
\]

\[
= \frac{(\prod_{j \in i} \sigma_j \models p_j) \cdot \sigma_i(x) \cdot p_i(x) \cdot (t \ll q_{i+1}(x))}{\sigma_1 \models q_1}.
\]

We now turn to the sum-increase setting of Lemma \[4.4.2\] in order to get a better version of both \( \sigma \) and \( t \). We take \( F : X \times \mathcal{D}(X)^k \times X^l \rightarrow \mathbb{R}_{\geq 0} \), to be:

\[
F(\sigma, t, \bar{x}) := \sigma(x_1) \cdot p_1(x_1) \cdot t(x_1)(x_2) \cdot p_2(x_2) \cdots t(x_{l-1})(x_l) \cdot p_l(x_l) = \sigma(x_1) \cdot p_1(x_1) \cdot \prod_{1 \leq k < l} t(x_k)(x_{k+1}) \cdot p_{k+1}(x_{k+1}). \tag{4.9}
\]

We also collect some results about \( F \).

**Lemma 4.5.3.** 1. \( F_i(\sigma, t) = (\sigma, t) \models \bar{p} \);
For each $1 \leq k < \ell$ one has:
\[
\sum_{x_1, \ldots, x_k} \sum_{x_{k+1}, \ldots, x_{\ell}} F(\sigma, t, \tilde{x}) = (\sigma_1 \models q_1) \cdot \gamma_k(x_k) \cdot t_{\tilde{q}_{k+1}}(x_{k+1}).
\]

**Proof.** 1 Since $F(\sigma, t) = \sum_{t} F(\sigma, t, \tilde{x}) = \sigma \models q_1 = (\sigma, t) \models \tilde{p}$.

2 Straightforward:
\[
\sum_{x_1, \ldots, x_k} \sum_{x_{k+1}, \ldots, x_{\ell}} F(\sigma, t, \tilde{x})
= \left( \sum_{x_1, \ldots, x_k} \sigma(x_1) \cdot p_1(x_1) \cdot t(x_1)(x_2) \cdot \cdots \cdot p_{k-1}(x_{k-1}) \cdot t(x_{k-1})(x_k) \right)
\cdot \sum_{x_{k+1}, \ldots, x_{\ell}} \left( \sum_{x_{k+2}, \ldots, x_{\ell}} t(x_{k+2})(x_{k+3}) \cdots p_{\ell}(x_\ell) \right)
= (\sigma_1 \models p_1) \cdots (\sigma_{k-1} \models p_{k-1}) \cdot \sigma_k(x_k)
\cdot \sum_{x_{k+1}, \ldots, x_{\ell}} \left( \sum_{x_{k+2}, \ldots, x_{\ell}} t(x_{k+1})(x_{k+2}) \cdots p_{\ell}(x_\ell) \right)
= (\sigma_1 \models q_1) \cdot \gamma_k(x_k) \cdot t_{\tilde{q}_{k+1}}(x_{k+1})
\]
by Lemma 4.5.2 (3)
\[
= (\sigma_1 \models q_1) \cdot \gamma_k(x_k) \cdot t_{\tilde{q}_{k+1}}(x_{k+1}).
\]

We now turn to the actual proof of the main result of this section.

**Proof.** (of Theorem 4.5.1) According to Lemma 4.4.2 we seek a better state and channel $\sigma', t'$ as maximum of the function $G(\sigma, t, -)$, where:
\[
G(\sigma, t, \sigma', t') = \sum_{x} F(\sigma, t, \tilde{x}) \cdot \ln(F(\sigma', t', \tilde{x})).
\]

Let $X = \{x_1, \ldots, x_n\}$. We write $w_i$ as variable for $\sigma'(x_i)$ and $z_{ij}$ for $t'(x_i)x_j)$. We use additional variables $\vec{k}$ and $\lambda$, as in Example 4.4.5, in a new function $H$, for the Lagrange multiplier method, where:
\[
H(\vec{w}, \vec{z}, \lambda, \vec{k})
= \sum_{x} F(\sigma, t, x) \cdot \ln \left( \prod_{1 \leq i \leq n} p_1(x_i) \cdot p_{k+1}(x_{k+1}) \right)
\cdot \left( (\sum_{i} w_i) - 1 \right)
\cdot \sum_{i} k_i \cdot \left( (\sum_{i} z_{i\ell}) - 1 \right)
= \sum_{x} F(\sigma, t, x) \cdot \ln \left( \prod_{1 \leq i \leq n} p_1(x_i) \cdot p_{k+1}(x_{k+1}) \right)
\cdot \left( (\sum_{i} w_i) - 1 \right)
\cdot \sum_{i} k_i \cdot \left( (\sum_{i} z_{i\ell}) - 1 \right).
\]
The partial derivatives of $H$ are as follows. First,
\[
\frac{\partial H}{\partial w_i}(\vec{w}, \vec{z}, \lambda, \vec{k})
= \sum_{x} F(\sigma, t, x) \cdot \frac{p_1(x_i)}{w_i} - \lambda = \frac{\sigma(x_i) \cdot q_1(x_i)}{w_i} - \lambda.
\]
Next, the expression for $z_{ij}$ can be rewritten using Lemma 4.5.2 (2).
\[
\frac{\partial H}{\partial z_{ij}}(\vec{w}, \vec{z}, \lambda, \vec{k})
= \sum_{x} F(\sigma, t, x) \cdot p_{k+1}(x_{k+1}) \cdot \frac{t_{\tilde{q}_{k+1}}(x_{k+1})}{z_{i\ell}} - k_i
\]
\[
= (\sigma_1 \models q_1) \cdot \sum_{1 \leq i \leq n} \gamma_i(x_i) \cdot t_{\tilde{q}_{k+1}}(x_{k+1}) - k_i.
\]
The partial derivatives w.r.t. the auxiliary variables are:

\[
\frac{\partial H}{\partial \lambda}(\vec{w}, \vec{z}, \lambda, \vec{\kappa}) = (\sum_i w_i) - 1 \\
\frac{\partial H}{\partial \kappa}(\vec{w}, \vec{z}, \lambda, \vec{\kappa}) = (\sum_j z_{ij}) - 1.
\]

When we set all these partial derivatives to zero, we can derive the following about \(\vec{w}\) and \(\vec{z}\). First we have:

\[
1 = \sum_i w_i = \sum_i \frac{\sigma(x_i) \cdot q_1(x_i)}{\lambda} = \sigma \overset{\lambda}{=} q_1.
\]

Hence \(\lambda = \sigma \overset{\lambda}{=} q_1\) and thus:

\[
w_i = \frac{\sigma(x_i) \cdot q_1(x_i)}{\lambda} = \frac{\sigma(x_i) \cdot q_1(x_i)}{\sigma \overset{\lambda}{=} q_1} = \sigma \overset{\lambda}{=} q_1(x_i) = \gamma_1(x_i).
\]

Thus we conclude that the better initial state \(\sigma'\) equals \(\gamma_1\), as in Theorem 4.5.1.

For the better channel \(t'\) we obtain, for each \(i\),

\[
1 = \sum_i' z_{ii'} = \sum_i' (\sigma_1 \overset{x_i'}{=} q_1) \cdot \sum_{1 \leq k < \ell} y_k(x_i) \cdot t_{q_i+1}(x_i)(x_{i'}).
\]

Hence \(\kappa_i = (\sigma_1 \overset{x_i}{=} q_1) \cdot \gamma(x_i)\), so that:

\[
z_{ii'} = \frac{(\sigma_1 \overset{x_i}{=} q_1) \cdot \sum_{1 \leq k < \ell} y_k(x_i) \cdot t_{q_i+1}(x_i)(x_{i'})}{\gamma(x_i)}
\]

This is the expression for \(t'\) in Theorem 4.5.1. The claimed increase in validity follows from Lemma 4.5.3[1].

\[\square\]

**Remark 4.5.4.** The description in Theorem 4.5.1 of the newly learned channel \(t'\), as convex sum of updated channels \(t_{q_i+1}\), is mathematically appealing, but is not optimal from a computational perspective: it involves several normalisations, namely for each channel update, and for the coefficients \(\gamma_i/\gamma\) of the convex sum. It is more efficient to simply multiply all factors \(\gamma_i(x), t(x)(y)\) and \(q_{i+1}(y)\) and then normalise at the very end. Getting into such implementation details is beyond the scope of this book.

An alternative way to compute the new transition channel \(t'\) is described in Exercise 4.5.2.
Chapter 4. Learning of States and Channels

We shall elaborate an example in the next section.

Exercises

4.5.1 Prove that in the situation of Subsection 4.5.2 one has:
\[ t_{q_{i+1}} \gg \gamma_i = \gamma_{i+1}. \]

*Hint:* Recall Exercise 2.5.5.

Conclude that there is a chain of morphisms in the category of kernels \( \text{Krn} \) of the form:
\[ (X, \gamma_1) \xrightarrow{\delta_n} (X, \gamma_1) \xrightarrow{\delta_n} \ldots \xrightarrow{\delta_n} (X, \gamma_{n+1}). \]

4.5.2 In the context of Theorem 4.5.1, consider a new sequence of joint states \( \tau_1, \ldots, \tau_\ell \in \mathcal{D}(X \times X) \), via:
\[ \tau_i \triangleq \langle (id, t) \gg \sigma_i \rangle \; |_{p \in q_{i+1}} \]
\[ = \langle id, n_{q_{i+1}} \rangle \gg (\sigma_i|_{p_i \& (i < q_{i+1})}) \text{ by Theorem 3.7.8} \]
\[ = \langle id, n_{q_{i+1}} \rangle \gg (\sigma_i|_{q_i}) \]
\[ = \langle id, n_{q_{i+1}} \rangle \gg \gamma_i. \]

Then define \( \tau \triangleq \sum_{1 \leq i \leq \ell} \tau_i \) and prove that the new channel \( t' \) in Theorem 4.5.1 can also be defined via disintegration as:
\[ t' = \tau[0,1][1,0]. \]

4.5.3 Write \( \omega_\ell = \langle id, t \rangle \gg \sigma \in \mathcal{D}(X^n \times X) \), for the joint state defined in (3.10) with \( e = id \). Prove that the states \( \gamma_i \in \mathcal{D}(X) \) in (4.8) arise as \( i \)-th marginal:
\[ \gamma_i = \omega_\ell |_{p_i \circ \cdots \circ p_0 \circ t_0 \cdots 0, \ldots, 0, 1, 0, \ldots, 0}, \]
with the (single) 1 in the mask on the \( i \)-th position, for \( 1 \leq i \leq \ell \), in the mask of length \( \ell + 1 \).

4.6 Learning hidden Markov models

In this section we extend the Baum-Welch algorithm from Markov chains to hidden Markov models (HMM). This means that we have to deal with an additional emmission channel. We shall see that the improvement approach for Markov chains in the previous section carries over HMMs. It only involves a bit more book-keeping. We shall concentrate on what is new, namely the
4.6. Learning hidden Markov models

emission channel. As before, we use a channel-based formulation and allow arbitrary factors as observables, and not just point observations.

After stating an proving the main result, we elaborate an example from the literature.

4.6.1 Finding a better hidden Markov model

Throughout this subsection we shall use:

• a fixed Hidden Markov model $\mathcal{H} = (1 \xrightarrow{\sigma'} X \xrightarrow{t} X \xrightarrow{e} Y)$;
• a list $\vec{p} = p_1, \ldots, p_\ell$ of factors on $Y$, from which we wish to learn;
• the filtered sequence of states $\sigma_1, \ldots, \sigma_{n+1} \in \mathcal{D}(X)$ with $\sigma_1 := \sigma$ and $\sigma_{i+1} := t \gg (\sigma'_{k<p_1})$, as introduced in Definition 3.4.4.

Our aim is to define a newly learned HMM $(\sigma', t', e')$.

We introduce two new sequences, like in the previous section. We shall use the same letters ($q$ and $\gamma$), but their definition is slightly different. In fact, the definitions given below coincide with the ones from (4.7) and (4.8) in the previous section where the emission channel $e$ is the identity $\text{id}$.

We start with a series of factors $q_1, \ldots, q_{\ell+1} \in \text{Fact}(X)$, defined in a ‘backward’ manner:

$$ q_{\ell+1} := 1 \quad \text{and} \quad q_{i-1} := (e \ll p_{i-1}) \& (t \ll q_i). \quad (4.10) $$

We can now define states $\gamma_i \in \mathcal{D}(X)$, for $1 \leq i \leq \ell$ via entry-wise updating, simply as:

$$ \gamma_i := \sigma_i \bigg|_{q_i} \quad \text{giving:} \quad \gamma := \sum_{1 \leq i < \ell} \gamma_i. \quad (4.11) $$

**Theorem 4.6.1.** In the above setting, starting from a Markov chain $(\sigma, t, e)$ we define the Baum-Welch improvement of $(\sigma, t, e)$ with factors $\vec{p} = p_1, \ldots, p_\ell$ as:

$$ \text{BW}(\sigma, t, e, \vec{p}) := (\sigma', t', e'), $$

where $\sigma' := \gamma_1$ and:

$$ t'(x) := \sum_{1 \leq i < \ell} \frac{\gamma_i(x)}{\gamma(x)} \cdot t_{q_i} \cdot (x) \quad \text{and} \quad e'(x) := \sum_{1 \leq i < \ell} \frac{\gamma_i(x)}{\gamma(x) + \gamma_i(x)} \cdot e_{p_i}\gamma^s(x). $$

Then: $\text{BW}(\sigma, t, e, \vec{p}) \models \vec{p} \geq (\sigma, t, e) \models \vec{p}$.

Notice that the new transition channel $t'$ is defined via a convex sum of $\ell - 1$ updated channels, whereas the new emission channel $e'$ involves $\ell$ summands. This makes sense, since $e$ can be updated via all $\ell$ factors $p_i$. Updating the transition channel $t$ proceeds via the pairs $(p_i, p_{i+1})$, of which there are only $\ell - 1$.  

235
For convenience, we overload the operation $BW$ and use it both for Markov chains (with three arguments) and for HMMs (with four arguments). It is not hard to see that the above newly learned state and transition channel $\sigma', t'$ satisfy:

$$(\sigma', t') = BW(\sigma, t, e \ll p_i),$$

with $BW$ as in Theorem 4.5.1.

We use Lemma 4.4.2 with function $F : \mathcal{D}(X) \times \mathcal{D}(X)^Y \times \mathcal{D}(Y)^X \times X' \times Y' \rightarrow \mathbb{R}_{\geq 0}$ defined by:

$$F(\sigma, t, e, \bar{x}, \bar{y}) := \sigma(x_1) \cdot e(x_1)(y_1) \cdot t(x_1)(x_2) \cdot e(x_2)(y_2) \cdot p_1(y_2) \cdots \cdot t(x_{i-1})(x_i) \cdot e(x_i)(y_i) \cdot p_1(y_i) \cdots \cdot t(x_{i-1})(x_i) \cdot e(x_i)(y_i) \cdot p_1(y_i) \cdots \cdot t(x_{i-1})(x_i) \cdot e(x_i)(y_i) \cdot p_1(y_i)$$

Then:

$$\sum_{\bar{x}, \bar{y}} F(\sigma, t, e, \bar{x}, \bar{y}) = \sum_{e} \sigma(x_1) \cdot t(x_1)(x_2) \cdot (e \ll p_1)(x_1) \cdots t(x_{i-1})(x_i) \cdot (e \ll p_1)(x_i)$$

$$= \sigma \models q_1$$

$$= (\sigma, t, e) \models \bar{p}.$$
for the new state $\sigma'$, transition channel $t'$ and emission channel $e'$. The function that we use for the Lagrange method is:

$$H(\bar{w}, \bar{z}, \bar{u}, \lambda, \vec{\kappa}, \vec{\mu}) := \sum_{i,j} F(\sigma, t, e, x_j, y_j) \cdot \ln \left( w_i \cdot u_{ij} \cdot p_1(y_{ij}) \cdot \prod_{1 \leq k < j} z_{ij,k} \cdot u_{ik,j} \cdot p_{k+1}(y_{ik}) \right)$$

\begin{align*}
&= \sum_{i,j} F(\sigma, t, e, x_j, y_j) \cdot \ln \left( w_i \cdot u_{ij} \cdot p_1(y_{ij}) \right) + \sum_{1 \leq k < j} \ln \left( z_{ij,k} \cdot u_{ik,j} \cdot p_{k+1}(y_{ik}) \right) - \lambda \cdot (\sum_i w_i) - 1 - \sum_i \kappa_i \cdot (\sum_k z_{ij,k}) - 1 - \sum_i \mu_i \cdot (\sum_j u_{ij}) - 1.
\end{align*}

As before, we take the partial derivatives of $H$, set them to zero, and solve the resulting equations in order to obtain the maximum. We shall only do so for the variables $u_{ij}$, corresponding to the new emission channel $e'$. The other cases work much as in the previous section. We use Lemma 4.6.2 to obtain:

$$\frac{\partial H}{\partial u_{ij}} = \sum_{i,j} F(\sigma, t, e, x_j, y_j) \cdot \ln \left( w_i \cdot u_{ij} \cdot p_1(y_{ij}) \right) + \sum_{1 \leq k < j} \ln \left( z_{ij,k} \cdot u_{ik,j} \cdot p_{k+1}(y_{ik}) \right) - \lambda \cdot (\sum_i w_i) - 1 - \sum_i \kappa_i \cdot (\sum_k z_{ij,k}) - 1 - \sum_i \mu_i \cdot (\sum_j u_{ij}) - 1.$$
This is the formula for $e$.

The sequence of observations that we wish to learn from is $l$. The predicates in a set $Y$.

There are three output elements: $e_{	ext{cola}}$, $e_{	ext{iced tea}}$, and $e_{	ext{lemon}}$.

We elaborate in detail the ‘crazy soft drink machine’ from [66, §9.2]. It has two internal positions $cp$ and $ip$ for ‘cola preference’ and ‘iced tea preference’, so $X = \{cp, ip\}$, with initial state:

$$\sigma = 1|cp).$$

The transition channel $t: X \rightarrow X$ is:

$$t(cp) = \frac{7}{10}|cp) + \frac{3}{10}|ip) \quad t(ip) = \frac{1}{2}|cp) + \frac{1}{2}|ip).$$

There are three output elements: $c$ for ‘cola’, $i$ for ‘iced tea’ and $l$ for ‘lemon’, in a set $Y = \{c, i, l\}$. The machine is called crazy because it may give a lemon drink, with some probability, in both internal positions, see the transition channel $e: X \rightarrow Y$.

$$e(cp) = \frac{6}{10}|c) + \frac{1}{10}|i) + \frac{3}{10}|l) \quad e(ip) = \frac{1}{10}|cp) + \frac{3}{10}|ip) + \frac{2}{10}|l).$$

The sequence of observations that we wish to learn from is $[l, i, c]$. It forms a sequence of point predicates $p_1 = 1_l, p_2 = 1_i, p_3 = 1_c$.

Pulling these predicates on $Y$ along $e: X \rightarrow Y$ yields:

$$(e \ll p_1)(cp) = \frac{1}{10} \quad (e \ll p_2)(cp) = \frac{1}{10} \quad (e \ll p_3)(cp) = \frac{6}{10} \quad (e \ll p_1)(ip) = \frac{2}{10} \quad (e \ll p_2)(ip) = \frac{7}{10} \quad (e \ll p_3)(ip) = \frac{1}{10}.$$

The sequence of filtered states starts with $\sigma_1 = \sigma = 1|cp)$ and continues with:

$$\sigma_2 = t \gg (\sigma_1|_{e \ll p_1}) = \frac{7}{10}|cp) + \frac{3}{10}|ip) \quad \sigma_3 = t \gg (\sigma_2|_{e \ll p_2}) = \frac{3}{10}|cp) + \frac{9}{20}|ip).$$

The predicates $q_1, q_2, q_3$ from [4.10] on $X$ satisfy $q_4 = 1$ and:

$$q_1(cp) = \frac{6}{10} \quad q_2(cp) = \frac{9}{200} \quad q_1(ip) = \frac{63}{200}$$
$$q_3(cp) = \frac{6}{10} \quad q_2(ip) = \frac{49}{200} \quad q_1(ip) = \frac{29}{1000}$$
In a similar way one obtains:

\[ \gamma_1 = 1|cp \]
\[ \gamma_2 = \frac{3}{10}|cp + \frac{7}{10}|ip \]
\[ \gamma_3 = \frac{22}{25}|cp + \frac{3}{25}|ip \]

The Baum-Welch learning step:

\[ \text{Baum-Welch learning step:} \]

channels:

\[ \text{two updated channels:} \]

\[ \gamma'(cp) = \frac{\gamma_1(cp)}{\gamma_1(cp) + \gamma_2(cp) + \gamma_3(cp)} \cdot I_{q_0} + \frac{\gamma_2(cp)}{\gamma_1(cp) + \gamma_2(cp) + \gamma_3(cp)} \cdot I_{q_1} + \frac{\gamma_3(cp)}{\gamma_1(cp) + \gamma_2(cp) + \gamma_3(cp)} \cdot I_{q_2} \]

The new emission channel \( e' \): \( X \rightarrow Y \) is described by the following convex sum of \( 3 \) updated channels:

\[ e'(cp) = \frac{\gamma_1(cp)}{\gamma_1(cp) + \gamma_2(cp) + \gamma_3(cp)} \cdot e_{p_1}(cp) + \frac{\gamma_2(cp)}{\gamma_1(cp) + \gamma_2(cp) + \gamma_3(cp)} \cdot e_{p_2}(cp) + \frac{\gamma_3(cp)}{\gamma_1(cp) + \gamma_2(cp) + \gamma_3(cp)} \cdot e_{p_3}(cp) \]

In a similar way one obtains:

\[ e'(ip) = \frac{6}{41}|c| + \frac{35}{41}|i| \approx 0.1463|c| + 0.8537|i| \]

The states \( \gamma_1 = \sigma_1|q_0 \) on \( X \) from (4.11) are:

\[ \gamma_1 = \sigma_1|q_0 \]
\[ \gamma_2 = \frac{3}{10}|q_2| + \frac{7}{10}|q_1 \]
\[ \gamma_3 = \frac{22}{25}|q_2| + \frac{3}{25}|q_1 \]

The new transition channel \( t' : X \rightarrow X \) is described by the following convex sum of \( 6 \) updated channels:

\[ t'(cp) \]

\[ = \frac{1}{1 + \frac{3}{10}/|cp|} \left( \frac{t(cp)(cp) \cdot q_2(cp)}{t \ll q_2(cp)} |cp| + \frac{t(cp)(ip) \cdot q_2(ip)}{t \ll q_2(cp)} |ip| \right) \]
\[ + \frac{3}{10}/|cp| \left( \frac{t(cp)(cp) \cdot q_1(cp)}{t \ll q_1(cp)} |cp| + \frac{t(cp)(ip) \cdot q_1(ip)}{t \ll q_1(cp)} |ip| \right) \]
\[ \approx \frac{1}{1 + \frac{3}{10}/|cp|} \left( \frac{t(cp)(cp)}{t \ll q_2(cp)} |cp| + \frac{t(cp)(ip)}{t \ll q_1(cp)} |ip| \right) \]
\[ \approx \frac{1}{1 + \frac{3}{10}/|cp|} \left( \frac{t(cp)(cp)}{t \ll q_2(cp)} |cp| + \frac{t(cp)(ip)}{t \ll q_1(cp)} |ip| \right) \]
\[ \approx \frac{29}{85}|cp| + \frac{49}{65}|ip| \approx 0.4462|cp| + 0.5538|ip| \]

In a similar way we get:

\[ t'(ip) = \frac{5}{11}|cp| + \frac{4}{11}|ip| \approx 0.8571|cp| + 0.1429|ip| \]

The new emission channel \( e' \): \( X \rightarrow Y \) is a convex sum of \( 3 \) updated channels:

\[ e'(cp) \]

\[ = \frac{1}{1 + \frac{3}{10}/|cp|} \left( \frac{e_{p_1}(cp)}{e_{p_1}(cp)} |cp| + \frac{e_{p_2}(cp)}{e_{p_2}(cp)} |ip| \right) \]
\[ = \frac{1}{1 + \frac{3}{10}/|cp|} \left( \frac{e_{p_1}(cp)}{e_{p_1}(cp)} |cp| + \frac{e_{p_2}(cp)}{e_{p_2}(cp)} |ip| \right) \]
\[ \approx \frac{4}{11}|c| + \frac{35}{41}|i| \approx 0.1463|c| + 0.8537|i| \]

In the end we like to point out the rise in HMM-validity as a result of the single Baum-Welch learning step:

\[ (\sigma, t, e) \models [p_1, p_2, p_3] = 0.0315 \]
\[ (\gamma_1, t', e') \models [p_1, p_2, p_3] = 0.0869 \]
After repeating the Baum–Welch steps ten times, this validity approaches \( \frac{1}{2} \).

**Remark 4.6.3.** As mentioned in the beginning of this subsection, the soft drink example is taken from [66] §9.3.3. The newly learned emission channel that is computed there coincides with the channel \( e' \) that we calculate above. But the new transition channel in [66] is different from the one that we obtain. The outcome of [66], written below as \( t' \), can be reconstructed as a convex sum of three updated channels, as in:

\[
\begin{align*}
t'(cp) &= \sum_{1 \leq i \leq 3} \frac{\gamma_i(cp)}{\sum_{1 \leq j \leq 3} \gamma_j(cp)} \cdot t|_{\mathbb{q}_1}(cp) = \frac{396}{585}(cp) + \frac{246}{585}|ip) \\
&\approx 0.548|cp) + 0.4514|ip) \\
t'(ip) &= \sum_{1 \leq i \leq 3} \frac{\gamma_i(ip)}{\sum_{1 \leq j \leq 3} \gamma_j(ip)} \cdot t|_{\mathbb{q}_1}(ip) = \frac{33}{41}|ip) + \frac{8}{37}|ip) \\
&\approx 0.6049|cp) + 0.1951|ip).
\end{align*}
\]

However, this does not fit the Baum-Welch steps as given in Theorem 4.6.1. Moreover, in this way we get a lower validity than when one uses a convex sum of two, as in the above calculation of the newly learned channel \( t' \), since:

\[
(\gamma_1, t', e') \models [p_1, p_2, p_3] = 0.0869 \quad (\gamma_1, t', e') \models [p_1, p_2, p_3] = 0.0724.
\]

### Exercises

**4.6.1** Prove in the general setting of Subsection 4.6.1:

\[
(\sigma', t, e) \models \bar{\rho} = \sigma_1 \models q_1 \leq \sigma|_{\mathbb{q}_1} \models q_1 = (\sigma', t, e) \models \bar{\rho}
\]

The first equation follows directly from (3.13).

**4.6.2** By repeatedly applying Baum-Welch to the crazy soda HMM \( \mathcal{H} \) from Subsection 4.6.2 we obtain an HMM \( \mathcal{H}^{\infty} = (\sigma^{\infty}, t^{\infty}, e^{\infty}) \) with \( \sigma^{\infty} = 1|cp) \) and with:

\[
\begin{align*}
r^{\infty}(cp) &= 1|ip) \\
e^{\infty}(cp) &= \frac{1}{2}|c) + \frac{1}{2}|l) \\
r^{\infty}(ip) &= 1|cp) \\
e^{\infty}(ip) &= 1|i).
\end{align*}
\]

Check that \( \mathcal{H}^{\infty} \models [1, 1, 1, 1] = \frac{1}{2} \).

**4.6.3** Compute the newly learned channel \( t' \) in Subsection 4.6.2 also via the formulas in Exercise 4.5.2 as disintegration of \( \tau = \tau_1 + \tau_2 + \tau_3 \) where \( \tau_i = ((\text{id}, t) \Rightarrow \sigma|)_{t|_{p|^{i}_{\gamma_0}}} \).

**4.6.4** This exercise gives an alternative way to the describe the newly learned emission channel, in the style of Exercise 4.5.2. In the context described in Subsection 4.6.1 define for \( 1 \leq i \leq \ell \),

\[
\rho_i := ((\epsilon, \text{id}) \Rightarrow \sigma|)_{t|_{p|^{i}_{\gamma_0}}} \in \mathcal{D}(\mathcal{Y} \times \mathcal{X})
\]
4.7 Data, validity, and learning

As we have described at the very beginning of this section, in learning one adapts one's state to given evidence, in the form of data. Specifically, one strives to increase the validity of the data. Although we have talked several times about data already, we have not yet made things precise. This is what this section will do: it will define what kind of data we’ll use, namely multisets of factors. It will then define validity of data in two fundamentally different ways, using either a 'multiple-state' perspective or a 'copied-state' perspective. This resembles the difference between drawing from an urn with replacement (multiple-state) or without replacement (copied-state). Finally the section ends with two learning theorems, one for each perspective. We shall see examples of both forms of learning in later sections.

4.7.1 Data and its validity

Data that is used for learning often comes in the form of sequences or (multidimensional) tables. The order of the data items does not matter for updating, see Lemma [2.3.5 (3)]. Thus the order is irrelevant, but multiple occurrences of data items do matter. These two aspects suggest that data items should be organised in the form of multisets, see Section [1.4]. Suppose we have a state \( \omega \in \mathcal{D}(X) \) on a set \( X \) and we wish to improve it in order to increase the validity of data on \( X \).

What form should this data have? In the light of what we just said, we would expect that data takes the form of a (natural) multiset \( \psi \in \mathcal{N}(X) \) on \( X \), with natural numbers as frequencies. But a recurring theme in this book is that instead of point predicates one can use more general fuzzy predicates or even factors. Indeed, in sections [4.5] and [4.6] we have described learning for Markov chains and hidden Markov models with data given by lists of factors. This is useful in cases when there is uncertainty about specific data items, or when data items are missing — so that a uniform predicate can be used, see Section [4.2]. Along those lines we should use multisets of factors as data.

**Definition 4.7.1.** Data on a set \( X \) means a multiset of factors \( \psi \in \mathcal{M}(\mathsf{Fact}(X)) \).
When such a $\psi$ contains only point point predicates $1_x$, for $x \in X$, we speak of point-data, and identify the data with a multiset $\psi \in M(X)$.

Notice that we do not require that we have natural multisets, with natural numbers of frequencies, even though in many examples and results that will be the case. We like to be as general as possible.

Next we wish to introduce validity for data. Before doing so we take a step back and look at some elementary issues. Assume we have a fair coin $\sigma = \frac{1}{2} |H\rangle + \frac{1}{2} |T\rangle$ as state. Consider the following questions.

1 What is the probability of getting two heads? Most people will immediately say $\frac{1}{4}$. The idea behind this answer is that we flip the coin twice, where getting a head has probability $\frac{1}{2}$ each time. More formally, this answer involves a ‘multiple-state’ perspective, where the coin state $\sigma$ is used twice, as in:

$$\sigma \otimes \sigma \models 1_H \otimes 1_H = (\sigma \models 1_H) \cdot (\sigma \models 1_H) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}.$$  

There is an alternative perspective in which the probability of getting two heads is $\frac{1}{2}$, namely: the coin is tossed once, and two (honest, truthful) observers are asked to tell what they see. They will both report ‘head’, so we get two heads, as required. This perspective is reflected by the computation:

$$\sigma \models 1_H \& 1_H = \sigma \models 1_H = \frac{1}{2}.$$  

2 We can similarly ask: what is the probability of getting head and tail? Let’s not worry about taking the order into account and simply ask about first seeing head, then tail. The first, multiple state perspective again gives $\frac{1}{4}$ as probability:

$$\sigma \otimes \sigma \models 1_H \otimes 1_T = (\sigma \models 1_H) \cdot (\sigma \models 1_T) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}.$$  

But if we toss the coin only once and look at the probability that the first of two observers reports ‘head’ and the second one reports ‘tail’ we get outcome zero:

$$\sigma \models 1_H \& 1_T = \sigma \models 0 = 0.$$  

3 Next assume it is dark and the observers cannot see the coin very clearly. We consider the predicate $p = 0.8 \cdot 1_H + 0.2 \cdot 1_T$ giving 80% certainty for head, and $q = 0.7 \cdot 1_H + 0.3 \cdot 1_T$ with only 70% certainty. What is now the probability of seeing $p$ and $q$? The multiple-state perspective gives:

$$\sigma \otimes \sigma \models p \otimes q = (\sigma \models p) \cdot (\sigma \models q)$$

$$= (\frac{1}{2} \cdot 0.8 + \frac{1}{2} \cdot 0.2) \cdot (\frac{1}{2} \cdot 0.7 + \frac{1}{2} \cdot 0.3) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}.$$  

242
4.7. Data, validity, and learning

The alternative perspective now gives:

$$
\sigma \models p \& q = \sigma \models \Delta \ll (p \otimes q) \quad \text{see Exercise 2.4.5(2)}
$$

In the latter line we have a copied state. As we already know from Subsection 1.8.2, in general $$\Delta \gg \sigma \not= \sigma \otimes \sigma$$. Hence these two perspectives are really different. This difference also played a role in our description of covariance and correlation in Section 2.9. These two perspectives also lead to different notions of validity of data.

**Definition 4.7.2.** Let $$\omega \in \mathcal{D}(X)$$ be a state and $$\psi \in \mathcal{M}(\text{Fact}(X))$$ be data on $$X$$.

1. The *multiple-state validity*, or M-validity for short, of $$\psi$$ in $$\omega$$ is defined as:

$$
\omega \models_M \psi := \prod_p (\omega \models p)^{\psi(p)}. \quad (4.12)
$$

2. The *copied-state validity*, or C-validity, of $$\psi$$ in $$\omega$$ is:

$$
\omega \models_C \psi := \omega \models_{\&} p^{\psi(p)}. \quad (4.13)
$$

When $$\psi$$ consists of point-data, these two validities become $$\prod_{x \in \text{supp}(\psi)} \omega(x)^{\psi(x)}$$ and $$\omega \models_{\&} \chi \models p(x)^{\psi(x)}$$, respectively.

For clarity: $$p^n$$ in point (2) is the $$n$$-fold conjunction $$p \& \cdots \& p$$, with $$p^0 = 1$$. When $$p$$ is sharp, then $$p^n = p$$, for $$n > 1$$. In particular, point-predicates $$1_x$$ are sharp. This formulation gives in general $$p^n(x) = p(x)^n$$. Similarly, $$r \in \mathbb{R}_{\geq 0}$$ we use $$p'(x) = p(x)^r$$, see also Exercise 4.7.4.

**Remark 4.7.3.** What the above definition ignores is the multiple counting of probabilities that needs to take place in order to handle different orders. For instance, if we ask about the probability of getting one head and two tails for a fair coin, we should take into account the (sum of the) probabilities for the three possible orders:

$$
H, T, T \quad T, H, T \quad T, T, H.
$$

This yields a probability of $$\frac{3}{8}$$, instead of $$\frac{1}{8}$$ given by $$\models_\otimes$$. 

243
More generally, for point-data $\psi = \sum_i n_i|x_i\rangle$ we should get as probability in state $\omega$:

$$\frac{(\sum_i n_i)!}{\prod_i n_i!} \prod_i \omega(x_i)^{n_i}.$$ 

Here we use the multinomial coefficient:

$$\binom{N}{\vec{n}} := \frac{N!}{n_1! \cdots n_k!}$$ 

where $N = n_1 + \cdots + n_k$.

It gives the number of ways to go from a multiset $\sum_i n_i|x_i\rangle$ to a list over $\{x_1, \ldots, x_k\}$. It was already used in Example 2.1.5.

In the current setting of learning we are interested in increasing validities. Then these normalisation constants do not play a role. That’s why they are omitted in the definitions of $\overrightarrow{M}$ and $\overrightarrow{C}$. Thus, to be precise, we should really refer to $\overrightarrow{M}$ and $\overrightarrow{C}$ as unnormalised validities.

### 4.7.2 Learning steps for data

We collect earlier results and organise them in the form of learning steps, both for the multiple-state and the copied-state perspective.

**Theorem 4.7.4.** Let $\omega \in \mathcal{D}(X)$ be a state, with data $\psi \in \mathcal{N} (\text{Fact}(X))$.

1. **Learning from a multiple-state perspective, or M-learning, can be done as follows.**

   $$\text{Mlrn}(\omega, \psi) \overrightarrow{M}\psi \geq \omega \overrightarrow{M}\psi \quad \text{for} \quad \text{Mlrn}(\omega, \psi) := \sum_p \frac{\psi(p)}{|\psi|} \cdot \omega_p,$$

   where $|\psi| = \sum_p \psi(p)$.

2. **Learning from a copied-state perspective, or C-learning, works via:**

   $$\text{Clrn}(\omega, \psi) \overrightarrow{C}\psi \geq \omega \overrightarrow{C}\psi \quad \text{for} \quad \text{Clrn}(\omega, \psi) := \omega_{k(p)}.$$ 

**Proof.**

1. This is a reformulation of Proposition 4.4.6(2).
2. Directly from Theorem 4.3.1(2). 

We have formulated M- and C-validity and M- and C-learning for natural multisets only, with natural numbers as frequencies. However, the above C-learning result works for arbitrary multisets. M-learning however depends on Proposition 4.4.6(2), which has been proven for natural numbers only.

Given data $\psi \in \mathcal{N}(\text{Fact}(X))$, the mappings

$$\omega \mapsto \text{Mlrn}(\omega, \psi) \quad \text{and} \quad \omega \mapsto \text{Clrn}(\omega, \psi)$$

in the above theorem form endofunctions $\mathcal{D}(X) \to \mathcal{D}(X)$. By iterating these
functions one may hope to find a fixed point. This is an important part of the learning process, which forms a topic in itself. It will not be investigated here. Instead, we concentrate on single learning steps.

When we apply point (1) to point-data \( \psi \in N(X) \) we get as better state:

\[
M_{\text{lrn}}(\omega,\psi) = \sum_x \frac{\psi(x)}{|\psi|} \omega(x) = \sum_x \frac{\psi(x)}{|\psi|} |x| = \text{Flrn}(\psi).
\]

Hence we rediscover frequentist learning \( \text{Flrn} \) from Section 1.6. The above result says that frequentist learning gives an increase of \( M \)-validity. In fact, it gives the highest possible validity. This is a standard result, see e.g. [61, Ex. 17.5], which we reproduce here.

**Proposition 4.7.5.** For non-empty point-data \( \psi \in \mathcal{M}(X) \) the predicate "\( M \)-validity of \( \psi \)"

\[
\mathcal{D}(X) \xrightarrow{(-) \ln \psi} \mathbb{R}_{\geq 0}
\]

takes its maximum at the distribution \( \text{Flrn}(\psi) \in \mathcal{D}(X) \) that is obtained by frequentist learning. Expressed differently, this means:

\[
\text{Flrn}(\psi) \in \arg\max_{\omega} \omega \ln \psi.
\]

**Proof.** Let \( \psi \) be a fixed non-empty multiset. We will seek the maximum of the function \( \omega \mapsto \omega \ln \psi \) by taking the derivative with respect to \( \omega \). We will work with the ‘log-validity’, that is, with the function \( \omega \mapsto \ln(\omega \ln \psi) \), where \( \ln \) is the monotone (natural) logarithm function. It reduces the product \( \prod \) of powers in the definition of \( \ln \) to a sum \( \sum \) of multiplications, as in Exercise 1.2.2.

Assume that the support of \( \psi = \sum_i r_i |x_i| \) is \( \{x_1, \ldots, x_n\} \subseteq X \). We look at distributions \( \omega \in \mathcal{D}([x_1, \ldots, x_n]) \); they may be identified with numbers \( v_1, \ldots, v_n \in \mathbb{R}_{\geq 0} \) with \( \sum_i v_i = 1 \). We thus seek the maximum of the log-validity function:

\[
k(\vec{v}) := \ln \left( \sum_i v_i |x_i| \right) = \ln \left( \prod_i v_i \right) = \sum_i r_i \cdot \ln(v_i).
\]

Since we have a constraint \( \sum_i v_i = 1 \) on the inputs, we can use the Lagrange multiplier method for finding the maximum. We thus take another parameter \( \lambda \) in a new function:

\[
K(\vec{v}, \lambda) := k(\vec{v}) - \lambda \cdot ((\sum_i v_i) - 1) = (\sum_i r_i \ln(v_i)) - \lambda \cdot ((\sum_i v_i) - 1).
\]

The partial derivatives of \( K \) are:

\[
\frac{\partial K}{\partial v_i}(\vec{v}, \lambda) = \frac{r_i}{v_i} - \lambda \quad \frac{\partial K}{\partial \lambda}(\vec{v}, \lambda) = 1 - \sum_i v_i.
\]
Chapter 4. Learning of States and Channels

Setting all of these to 0 and solving gives the required maximum:

\[ 1 = \sum_i v_i = \sum_i \frac{r_i}{\lambda} = \frac{\sum_i r_i}{\lambda}. \]

Hence \( \lambda = \sum_i r_i \) and thus:

\[ v_i = \frac{r_i}{\lambda} = \frac{r_i}{\sum_i r_i} \text{ Flrn}(\psi)(x_i). \]

Informally, this proposition allows us to say that \( \text{Flrn}(\psi) \) is the distribution that best fits the data \( \psi \in \mathcal{M}(X) \). Alternatively, there is an inequality between \( \mathcal{M} \)-validities:

\[ \omega \models \psi \leq \text{Flrn}(\psi) \models \psi, \quad \text{for each } \omega \in D(X). \]

Notice that Proposition 4.7.5 requires that \( \psi \) consists of point-data. For non-point-data there is no such maximality result, see Exercise 4.7.3 below.

We conclude this section with some additional observations about frequentist learning. We first show how frequentist learning can be done by conditioning a uniform state.

**Proposition 4.7.6.** Let multiset \( \varphi \in \mathcal{M}(X) \) form point-data on \( X \) with (finite, non-empty) support \( S \subseteq X \). We consider \( \varphi \) as a factor, see Remark 2.1.6. Then:

1. \( \text{Flrn}(\varphi) = \upsilon|_{S} \), where \( \upsilon \) is the uniform distribution on \( S \);
2. \( \text{Flrn}(r \cdot \varphi) = \text{Flrn}(\varphi) \), for \( r \in \mathbb{R}_{>0} \);
3. \( \text{Flrn}(\varphi \& \psi) = \text{Flrn}(\varphi)|_{\psi} \).

**Proof.** 1. Let the support \( S \) of \( \varphi \) have \( n \) elements, so that \( \upsilon = \sum_{x \in S} \frac{1}{n} |_{x} \). We simply follow the definition of conditioning:

\[ \upsilon|_{S}(x) = \frac{\upsilon(x) \cdot \varphi(x)}{\upsilon \models \varphi} = \frac{1/n \cdot \varphi(x)}{\sum_{y \in S} 1/n \cdot \varphi(y)} = \frac{\varphi(x)}{\sum_{y \in S} \varphi(y)} = \text{Flrn}(\varphi)(x). \]

2. Obvious.
3. We use the point (1) twice and Lemma 2.3.5 in:

\[ \text{Flrn}(\varphi \& \psi) = \upsilon|_{S \& \psi} = \upsilon|_{S \psi} = \text{Flrn}(\varphi)|_{\psi}. \]

The last point gives a multiplicative compositionality result: in order to learn from the (pointwise) product of two multisets, we can learn from them successively. This is nice, but it would be more useful to have an additive compositionality result, involving a sum of multisets, as in \( \text{Flrn}(\varphi + \psi) \), since then we can add new data to what we have already learnt from an existing table (multiset) of data. Such an additive learning result will appear later in Proposition 4.8.7.

We conclude with a couple of additional points.
4.7. Data, validity, and learning

Remark 4.7.7. Suppose we have a space \( Y = \{y_1, \ldots, y_n\} \) and a multiset \( \psi = \sum_i n_i y_i \) of data over \( Y \). If \( n_i = 0 \) for some \( i \), then also \( \text{Flrn}(\psi|y_i) = 0 \). This is sometimes undesirable, since such a zero value remains zero in subsequent updates. This situation is referred to as the zero count problem.

A common solution is to use Laplacian learning \( \text{Llrn} : \mathcal{M}(X) \rightarrow \mathcal{D}(X) \), which is defined as:

\[
\text{Llrn}(\psi) \coloneqq \text{Flrn}(\psi + 1),
\]

where \( 1 \in \mathcal{M}(X) \) is the multiset \( \sum_i 1|x_i \) in which each element occurs once.

Thus, for \( n = 3 \) and \( \psi = 3|y_1 \} + 2|y_3 \} \) we get:

\[
\text{Flrn}(\psi) = \frac{1}{2}|y_1 \} + \frac{2}{3}|y_3 \}
\]

whereas \( \text{Llrn}(\psi) = \frac{1}{2}|y_1 \} + \frac{1}{3}|y_3 \} + \frac{1}{3}|y_3 \} \).

Remark 4.7.8. At the beginning of this section we suggested that the multiple/copied state difference is comparable to the familiar urn-model with/without replacement. We like to make this a bit more precise now.

Suppose that we have a state \( \omega \) and we have a sequence of three predicates \( p_1, p_2, p_3 \) whose validity we like to determine consecutively, without replacement. This means that after each observation we update the state:

1. We determine \( \omega \models p_1 \) and proceed to the next, updated state \( \omega|p_1 \).
2. We determine \( \omega|p_1 \models p_2 \) in the updated state, and proceed to the next level update with \( \omega|p_1, p_2 \models p_3 \), see Lemma 2.3.5 (3).
3. We determine \( \omega|p_1, p_2 \models p_3 \).

The combined validity obtained after these steps can be rewritten via Bayes’ product rule — see Proposition 2.3.3(1) — as copied-state validity:

\[
(\omega \models p_1) \cdot (\omega|p_1 \models p_2) \cdot (\omega|p_1, p_2 \models p_3)
= (\omega \models p_1 \land p_2) \cdot (\omega|p_1, p_2 \models p_3)
\begin{align*}
&= \omega \models p_1 \land p_2 \land p_3 \\
&= \omega \models 1|p_1 \} + 1|p_2 \} + 1|p_3 \}.
\end{align*}
\]

Exercises

4.7.1 Check the following M-validities of point-data:

\[
\frac{1}{8}|H \} + \frac{1}{4}|T \} \models 1|H \} + 2|T \} = \frac{1}{8} \\
\frac{1}{9}|H \} + \frac{2}{3}|T \} \models 1|H \} + 2|T \} = \frac{1}{9} \\
\frac{1}{3}|H \} + \frac{2}{3}|T \} \models 1|H \} + 2|T \} = \frac{7}{12}.
\]

Conclude that in general, for an arbitrary multiset \( \psi \in \mathcal{M}(X) \), the operation \( (\cdot) \models 1 \models \psi : \mathcal{D}(X) \rightarrow [0, 1] \) does not preserve convex sums of states.
4.7.2 Consider the channel $c: \{H, T\} \to \{0, 1, 2\}$ defined by:
\[
c(H) = \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|2\rangle \quad c(T) = \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|2\rangle,
\]
with state $\sigma = \frac{1}{2}|H\rangle + \frac{1}{2}|T\rangle$ and point-data $\psi = 3|1\rangle + 6|2\rangle$. Check that:
1. $c \gg \sigma = \frac{5}{32}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|2\rangle$;
2. $c \ll \psi = 3|H\rangle + 2|T\rangle$;
3. $c \gg \sigma \ll \psi = \frac{10592}{24200}$;
4. $\sigma \ll c \ll \psi = \frac{1}{32}$.

We conclude that $c \gg \sigma \ll \psi$ and $\sigma \ll c \ll \psi$ are different — whereas the analogues for validity $\models$ are the same, see Proposition 2.4.3. As we shall see later in Exercise 4.8.5, there is more to say.

4.7.3 In Proposition 4.7.5 we have seen that for point-data $\psi \in \mathcal{M}(X)$ the function $(-) \ll \psi: D(X) \to \mathbb{R}_{\geq 0}$ takes its maximum at $\operatorname{Flrn}(\psi) = \operatorname{Mlrn}(\omega, \psi)$. This exercise demonstrates that there is no analogous maximality for non-point-data $\varphi \in \mathcal{M}(\operatorname{Fact}(X))$.

Take $X = \{0, 1\}$ with predicates:
\[
p_1 = \frac{1}{2} \cdot 1_0 + \frac{1}{2} \cdot 1_1 \quad p_2 = 1 \cdot 1_0 + \frac{1}{2} \cdot 1_1.
\]
We combined them into non-point-data $\varphi = 1|p_1\rangle + 1|p_2\rangle \in \mathcal{M}(\operatorname{Fact}(X))$.

Consider the two states:
\[
\omega = \frac{1}{3}|0\rangle + \frac{2}{3}|1\rangle \quad \sigma = \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle
\]
Check that:
1. $\omega \ll \varphi = \frac{55}{128} \approx 0.43$;
2. $\operatorname{Mlrn}(\omega, \varphi) = \frac{65}{64}|0\rangle + \frac{99}{64}|1\rangle$;
3. $\operatorname{Mlrn}(\omega, \varphi) \ll \varphi \approx \frac{10579}{24200} \approx 0.437$;
4. $\sigma \ll \varphi = \frac{55}{128} \approx 0.469$.

4.7.4 1 Prove that for each set $X$, the collection $\operatorname{Fact}(X)$ of factors $X \to \mathbb{R}_{\geq 0}$ on $X$ is a multiplicative cone, via its multiplicative commutative monoid structure $(1, \&)$ and its power scalar multiplication $\mathbb{R}_{\geq 0} \times \operatorname{Fact}(X) \to \operatorname{Fact}(X)$ given by $(s, p) \mapsto p^s$, where $p^s(x) = p(x)^s$.

2 Consider the ‘curried’ version of M-validity as a function:

\[
\mathcal{M}(\operatorname{Fact}(X)) \xrightarrow{L} \operatorname{Fact}(\mathcal{D}(X)) \text{ given by } L(\psi)(\omega) := \omega \ll \psi.
\]

Prove that $L$ is a morphism of cones, from multisets with their additive cone structure, to factors with their multiplicative cone structure.
4.7. Data, validity, and learning

Concretely, this amounts to:
\[ \omega \models (\varphi + \psi) = (\omega \models \varphi) \cdot (\omega \models \psi) \]
\[ \omega \models \varphi = 1 \]
\[ \omega \models (s \cdot \varphi) = (\omega \models \varphi)^s. \]

3 Conclude that \( L \) is the free extension — see Exercise 1.4.9 (2) — of the ‘factor-validity’ function \( \text{fv}: \text{Fact}(X) \rightarrow \text{Fact}(D(X)) \) given by:
\[ \text{fv}(p)(\omega) := \omega \models p. \]

4.7.5 In Proposition 4.7.5 we have seen how the state \( \text{Flrn}(\psi) \) learned from a multiset \( \psi \in M(X) \) gives maximal M-validity \( \models \) to \( \psi \). There is an alternative way to describe \( \text{Flrn}(\psi) \) as maximal, using ordinary validity \( \models \) (plus entropy) instead of M-validity, via a formulation from \([76]\). It involves turning the multiset \( \psi \in M(X) \) into an observable \( \ln \psi : X \rightarrow \mathbb{R} \) given by \( (\ln \psi)(x) = \ln (\psi(x)) \).

The claim is:
\[ \text{Flrn}(\psi) \in \arg\max_{\omega} (\omega \models \ln \psi + H(\omega)), \]
where \( H(\omega) \) is the Shannon entropy, see Definition 2.1.3 (1), expressed via the natural logarithm \( \ln \).

1 Check that:
\[ \omega \models \ln \psi + H(\omega) = \omega \models (\psi/\omega). \]

2 Write \( X = \{x_1, \ldots, x_n\} \) and use variable \( w_i \) for \( \omega(x_i) \). Consider the Lagrangian:
\[ F(\vec{w}, \lambda) := \sum_i w_i \cdot \ln (\psi(x_i)/w_i) - \lambda \cdot ((\sum_i w_i) - 1). \]

Check that its partial derivatives are:
\[ \frac{\partial F}{\partial w_i}(\vec{w}, \lambda) = \ln (\psi(x_i)/w_i) - 1 - \lambda \quad \frac{\partial F}{\partial \lambda}(\vec{w}, \lambda) = (\sum_i w_i) - 1 \]

3 Set these partial derivatives to zero and derive:
\[ w_i = \frac{\psi(x_i)}{e^{\lambda+1}} \quad \text{and} \quad e^{\lambda+1} = |\psi| = \sum_i \psi(x_i). \]

Conclude that \( w_i = \text{Flrn}(\psi)(x_i) \).
Chapter 4. Learning of States and Channels

4.8 Learning along a channel

In the previous section we have looked at validity of data in a state and how to increase this validity by adapting the state. So far we have considered the situation where the state and data are on the same set \( X \). In practice, it often happens that there is a difference, like in:

\[
\begin{array}{ccc}
\pi & \xrightarrow{e} & \pi' \\
\text{state to be learned} & & \text{data}
\end{array}
\]  

(4.15)

We will assume that there is a channel between the two spaces — as in the above picture — that can be used to mediate between the given data and the state that we wish to learn. This is what we call ‘learning along a channel’. This learning challenge is often described in terms of ‘hidden’ or ‘latent’ variables, since the elements of the space \( X \) are not directly accessible, but only indirectly via the ‘emmission’ channel \( e \). In later chapters we shall deal with a different situation, where the channel \( e \) becomes a learning goal in itself. But for now, we assume that it is given.

We describe two different ways to handle this new learning situation along a channel, depending on whether one takes the multiple-state or the copied-state perspective. In both cases we first move data \( \psi \in \mathcal{M}(\text{Fact}(Y)) \) on \( Y \) to data on \( X \), via the channel \( e: X \to Y \). This can be done easily, by using that the multiset operation \( \mathcal{M} \) is functorial — works not only on sets but also on functions between them, see Definition 1.4.1 — and that factor transformation yields a function \( e \ll (-): \text{Fact}(Y) \to \text{Fact}(X) \). Thus we get data on \( X \) via a new data transformation, which we shall write as:

\[
e \ll \mathcal{M} \psi := \mathcal{M}(e \ll (-))(\psi) = \sum_p \psi(p) \cdot e \ll p.
\]

(4.16)

**Definition 4.8.1.** Let \( e: X \to Y \) be a channel with data \( \psi \in \mathcal{M}(\text{Fact}(Y)) \) on its codomain. We define \( \mathcal{M}/C \)-learning along \( e \) from \( \psi \) as \( \mathcal{M}/C \)-learning from \( e \ll \mathcal{M} \psi \) in (4.16). Concretely this turns a state \( \omega \in \mathcal{D}(X) \) into a newly learned states on \( X \), where:

1 for \( \mathcal{M} \)-learning,

\[
\mathcal{M}\text{lrn}(\omega, e, \psi) := \mathcal{M}\text{lrn}(\omega, e \ll \mathcal{M} \psi) = \sum_p \frac{\psi(p)}{|\psi|} \cdot \omega|_{e \ll p}.
\]

(4.17)

2 for \( C \)-learning,

\[
\mathcal{C}\text{lrn}(\omega, e, \psi) := \mathcal{C}\text{lrn}(\omega, e \ll \mathcal{M} \psi)
\]

\[
= \omega|_{\bigcup_{i \in p \cap \mathcal{N}(e \ll \mathcal{M} \psi)}}
\]

\[
= \omega|_{\bigcup_{i \in p \cap \mathcal{N}(e \ll \mathcal{M} \psi)}} \quad \text{when } \psi = \sum_{1 \leq i \leq k} n_i|_{p_i}.
\]

(4.18)
4.8. Learning along a channel

Both formulations $\text{Mlrn}(\omega, e, \psi)$ and $\text{Clrn}(\omega, e, \psi)$ for the newly learned states come directly from Theorem 4.7.4 using data of the form (4.16). They guarantee a higher M/C-validity for the transformed data:

- $\text{Mlrn}(\omega, e, \psi) \parallel e \ll_M \psi \geq \omega \parallel e \ll_M \psi$
- $\text{Clrn}(\omega, e, \psi) \parallel e \ll_M \psi \geq \omega \parallel e \ll_M \psi$

The last equation in (4.18) follows from Lemma 2.3.5 (3). Notice that we have overloaded the notation $\text{Mlrn}$ and $\text{Clrn}$, where its meaning depends on the number of arguments: two for ordinary learning and three for learning along a channel.

Much of the remainder of this section will be devoted to examples from the literature. The M/C distinction is made here, following [48], and is not part of the description of these examples in the literature.

We start with M-learning along a channel. We first point out that in the special case where we have point-data, M-learning along a channel in (4.17) can be expressed via a dagger channel, as Jeffrey’s adaptation (see Section 3.11).

**Lemma 4.8.2.** For a state $\omega \in \mathcal{D}(X)$ and channel $e : X \to Y$ with point-data $\varphi \in \mathcal{M}(X)$ we get:

$$\text{Mlrn}(\omega, e, \varphi) = e^\dagger \omega \gg \text{Flrn}(\varphi). \quad (4.19)$$

**Proof.** Since:

1. $\text{Mlrn}(\omega, e, \varphi) = \sum_x \frac{\varphi(x)}{[\varphi]} \cdot \omega_{[e]}$
2. $= \sum_x \frac{\varphi(x)}{[\varphi]} \cdot e^\dagger_{[e]}(x)$ by Proposition 3.7.2
3. $= \sum_x \text{Flrn}(\varphi)(x) \cdot e^\dagger_{[e]}(x)$
4. $= e^\dagger_{[e]} \gg \text{Flrn}(\varphi)$ by Exercise 1.7.3 (2).

This special case occurs in the next two illustrations.

**Example 4.8.3.** Consider the following Bayesian network from [88 §20.3].

It involves two bags, labelled 0 and 1, containing candies. These candies can
have one of two flavours, cherry \((C)\) or lime \((L)\), with a red \((R)\) or green \((G)\) wrapper, and with a hole \((H)\) or not \((H^\perp)\).

The network \((4.20)\) involves three channels, written as \(f: \{0,1\} \rightarrow \{C,L\}\), \(w: \{0,1\} \rightarrow \{R,G\}\), \(h: \{0,1\} \rightarrow \{H,H^\perp\}\). They have equal probabilities in [88]:

\[
\begin{align*}
    f(0) &= \frac{6}{10}\ket{C} + \frac{4}{10}\ket{L} \\
    f(1) &= \frac{4}{10}\ket{C} + \frac{6}{10}\ket{L} \\
    w(0) &= \frac{9}{10}\ket{R} + \frac{1}{10}\ket{G} \\
    w(1) &= \frac{1}{10}\ket{R} + \frac{9}{10}\ket{G} \\
    h(0) &= \frac{6}{10}\ket{H} + \frac{4}{10}\ket{H^\perp} \\
    h(1) &= \frac{4}{10}\ket{H} + \frac{6}{10}\ket{H^\perp}.
\end{align*}
\]

Here we combine these three channels into a single 3-tuple channel \((f, w, h) = (f \otimes w \otimes h) \circ \Delta_3: \{0,1\} \rightarrow \{C,L\} \times \{R,G\} \times \{H,H^\perp\}\), as in \((4.20)\). On input 0 this tuple is:

\[
(f, w, h)(0) = f(0) \otimes w(0) \otimes h(0) = \frac{216}{1000}\ket{C,R,H} + \frac{144}{1000}\ket{C,R,H^\perp} + \frac{144}{1000}\ket{C,G,H} + \frac{96}{1000}\ket{C,G,H^\perp} + \frac{144}{1000}\ket{L,R,H} + \frac{96}{1000}\ket{L,R,H^\perp} + \frac{96}{1000}\ket{L,R,H} + \frac{64}{1000}\ket{L,R,H^\perp}.
\]

The data to learn from in [88] involves 1000 data elements in total, about combinations of flavours, wrappers and holes. From these data we like to learn a distribution on bags, giving the probabilities that the candies that produce these data come from bag 0 or 1. Here we formalise the data as a multiset \(\psi \in \mathcal{M}(\{C,L\} \times \{R,G\} \times \{H,H^\perp\})\) with the following multiplicities:

\[
\psi = 273\ket{C,R,H} + 93\ket{C,R,H^\perp} + 104\ket{C,G,H} + 90\ket{C,G,H^\perp} + 79\ket{L,R,H} + 100\ket{L,R,H^\perp} + 94\ket{L,R,H} + 167\ket{L,R,H^\perp}.
\]

We are now set for M-learning along the tuple channel \((f, w, h)\) from these point-data on its codomain, via the dagger of the tuple channel, see Lemma 4.8.2. In [88] this uses a prior distribution \(\rho = \frac{6}{10}\ket{0} + \frac{4}{10}\ket{1}\). We thus compute the newly learned distribution on \([0,1]\) via Jeffrey’s rule:

\[
\text{Mlrn}(\rho, \langle f, w, h \rangle, \psi)^{1000} \gg \text{Flrn}(\psi) = \frac{1000}{30091}\ket{0} + \frac{10540}{30091}\ket{1} \approx 0.6124\ket{0} + 0.3876\ket{1}.
\]

This probability 0.6124 is exactly as computed in [88] §20.3, but without any of the channel and dagger machinery. Moreover, what we call the ‘multiple-state’ perspective is not made explicit there.

We add another example, which combines M-learning and C-learning along a channel.

**Example 4.8.4.** The following is a classical data analysis challenge from [86] that is used for instance in [21] to motivate the Expectation-Maximisation algorithm (see Section 4.9 below for more information). We shall describe it first

252
as M-learning along a channel, as direct translation of the description in [21].

The channel that we use looks much like the one in the Bayesian learning of a coin-bias in Example 2.6.2.

The example involves point-data on four kinds of animals, which we shall simply categorise via the 4-element set 4 = \{0, 1, 2, 3\}. The point-data is ψ = 125|0⟩ + 18|1⟩ + 20|2⟩ + 34|3⟩ ∈ M(4). A (fixed) channel e : [0, 1] → 4 is given with the unit interval as sample space, and:

\[ e(r) = \left( \frac{1}{2} + \frac{1}{4} \cdot r \right)|0⟩ + \frac{1}{2} \cdot (1 - r)|1⟩ + \frac{1}{4} \cdot (1 - r)|2⟩ + \frac{1}{4} \cdot r|3⟩ \]

The aim is to find the value \( r \in [0, 1] \) that best fits the data ψ.

We will discretise the unit interval, like in Section 2.6 and chop it up in \( N = 100 \) parts, giving as subspace \([0, 1]_N \hookrightarrow [0, 1]\), see Definition 2.6.1, so we can consider \( e \) as a channel \( e : [0, 1]_N \rightarrow 4 \). We start without any assumption, via the uniform distribution \( ω_0 = υ \) on \([0, 1]_N\).

We can now successively learn new distributions on \([0, 1]_N\) via:

\[ ω_{i+1} = Clrn(ω_i, e, ψ) \]

After running these M-learning steps 100 times we get the distribution in the bar chart on the left in Figure 4.1. Its expected value is 0.6268214581, which differs only slightly from the computed value, see Exercise 4.8.3 (copied from [21]).

With the knowledge that we have accumulated by now about learning it is good to look back at our earlier approach to coin bias learning in Example 2.6.2. There we had a channel \( Flip : [0, 1]_N \rightarrow 2 \) where 2 = \{0, 1\} is used for head (1) and tail (0). We had a list of data items \([0, 1, 1, 1, 0, 0, 1, 1]\) from which we wished to learn the coin bias in \([0, 1]\). We would now consider the data not as a list but as a multiset \( ϕ = \frac{3}{1}|0⟩ + \frac{5}{1}|1⟩ \). In Example 2.6.2 we (also) started from a uniform prior \( υ \) on \([0, 1]_N\) and performed eight successive conditionings, following the data \([0, 1, 1, 1, 0, 0, 1, 1]\)

\[ υ_{[l≤c_i]} = Clrn(υ_{[l≤c_i]}, ϕ) \]

Thus, without knowing at the time, we already saw an example of C-learning along a channel in Example 2.6.2.

But having seen this, we can ask: what do we get if we apply C-learning in the original example from [86] that we started from. Doing 100 steps as above, but now with \( ω_{i+1} = Clrn(ω_i, e, ψ) \), gives the bar chart on the right in Figure 4.1. Since this involves a focused spike, we have use \( N = 1000 \) in the discretisation in order to get a clearer picture. The expected value is now a little bit different, namely: 0.6267803732. Thus, the distributions obtained via
M- or C-learning along a channel are radically different, but not their expected values.

We include a somewhat similar example, but use it to illustrate another difference between M-learning and C-learning, namely the number of iterations involved. Since this is only one example, without a general analysis, everything we can conclude from it is that the number of iterations to reach a certain level of stability may vary considerably.

**Example 4.8.5.** We analyse an image-processing illustration from [74] which can be structured clearly in terms of two different channels, namely of the form:

\[
[0, 1] \overset{e}{\rightarrow} 3 \overset{d}{\rightarrow} 2
\]

The set \(3 = \{0, 1, 2\}\) refers to three forms of shapes (round dark, square dark, light), which are mapped to the set \(2 = \{0, 1\}\) for only dark and light. The channel \(d: 3 \rightarrow 2\) is deterministic, with \(d(0) = d(1) = 1|0\) and \(d(2) = 1|1\).

The channel \(e: [0, 1] \rightarrow 3\) is given, following [74], as:

\[
e(r) := \frac{1}{3}|0\rangle + \frac{1}{3}(1 + r)|1\rangle + \frac{1}{3}(2 - r)|2\rangle.
\]

Like in Example 4.8.4 we discretise the unit interval \([0, 1]\) into \([0, 1]_N\) for \(N = 100\), with uniform prior state \(\omega_0 = \nu\) on \([0, 1]_N\).

The data used in [74] involve 63 dark objects. The aim is to learn how many of them are round or square. In this set-up, light objects are mapped to light objects by the channel \(d\), via \(d(2) = 1|1\), so the number of light objects in the data does not matter very much. We choose 37, in order to get a round number of 100 data items in total. Thus we have point-date \(\psi = 63|0\rangle + 37|1\rangle \in \mathcal{N}(Y)\).

The computations in [74] Table 1 quickly lead to a division of the 63 dark objects into 25 square and 38 round, for a learned value of \(r = 0.520000 \in\)
4.8. Learning along a channel

[0, 1]. In this computation an a priori equal division of 63 is somehow assumed with \( r = 0.379562 \). Here we shall approximate these outcomes in a slightly different way, starting from only a uniform prior on \([0, 1]_N\).

What we first need to determine is along which channel to learn in Diagram 4.21. We have a prior \( \omega_0 \) on \([0, 1]_N \subseteq [0, 1]\) and point-data \( \psi \) on \( Y \), so we should learn along the composite channel \( d \circ e : [0, 1]_N \rightarrow Y \). In M-form this proceeds via:

\[
\omega_{i+1} := M_{\text{lrn}}(\omega_i, d \circ e, \psi) \quad \text{(4.19)}
\]

Recall that the point of this example is to learn the actual division of square dark objects (represented by the element 0 in the set 3) and round dark objects (represented by 1 in 3). In order to obtain this division we do state transformation in Diagram 4.21 along \( e : [0, 1] \rightarrow 3 \), in order to obtain the distribution \( e \circ \omega_i \) on the set 3. In addition, we like to learn the expected value (mean) of \( \omega_i \). The following table gives an impression.

<table>
<thead>
<tr>
<th>M-learning steps</th>
<th>( e \gg \omega_i )</th>
<th>mean(( \omega_i ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i = 0 )</td>
<td>( 0.25</td>
<td>0\rangle + 0.3738</td>
</tr>
<tr>
<td>( i = 10 )</td>
<td>( 0.25</td>
<td>0\rangle + 0.375</td>
</tr>
<tr>
<td>( i = 25 )</td>
<td>( 0.25</td>
<td>0\rangle + 0.3764</td>
</tr>
<tr>
<td>( i = 100 )</td>
<td>( 0.25</td>
<td>0\rangle + 0.3793</td>
</tr>
<tr>
<td>( i = 250 )</td>
<td>( 0.25</td>
<td>0\rangle + 0.38</td>
</tr>
</tbody>
</table>

The last line reflects the outcomes in [74].

We now do the same experiment with C-learning along the channel \( d \circ e \), so that we now take \( \omega_{i+1} := C_{\text{lrn}}(\omega_i, d \circ e, \psi) \). Then we get the following table:

<table>
<thead>
<tr>
<th>C-learning steps</th>
<th>( e \gg \omega_i )</th>
<th>mean(( \omega_i ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i = 0 )</td>
<td>( 0.25</td>
<td>0\rangle + 0.3738</td>
</tr>
<tr>
<td>( i = 10 )</td>
<td>( 0.25</td>
<td>0\rangle + 0.3797</td>
</tr>
<tr>
<td>( i = 25 )</td>
<td>( 0.25</td>
<td>0\rangle + 0.3799</td>
</tr>
<tr>
<td>( i = 60 )</td>
<td>( 0.25</td>
<td>0\rangle + 0.38</td>
</tr>
</tbody>
</table>

We thus see that in this example, C-learning along a channel requires in the order of 4-5 times fewer steps than M-learning. The resulting distributions on
Figure 4.2 Distribution obtained after 250 M-learning steps (on the left) and after 60 C-learning steps (on the right) in Example 4.8.5 on the (discretised) unit interval $[0, 1]$.

$[0, 1]$ are given as bar charts in Figure 4.2. They have the same shapes as in Figure 4.1.

We turn to another example, also with a comparison between M-learning and C-learning.

**Example 4.8.6.** Reference [27] explains Expectation-Maximisation in computational biology, mainly by providing an example. It involves classification of a coin in one of two classes, depending on its bias. There are five separate sets of data (multisets), each with ten coin outcomes head ($H$) and tail ($T$), see the first column in Table (4.22) below. There is a channel $e : \{0, 1\} \rightarrow \{H, T\}$ that captures two coins, with slightly different biases:

$$e(0) = \frac{3}{5}|H\rangle + \frac{2}{5}|T\rangle \quad \text{and} \quad e(1) = \frac{1}{2}|H\rangle + \frac{1}{2}|T\rangle.$$

Thus, the distribution $e(0)$ shows a slight bias towards head, whereas $e(1)$ is unbiased. By learning along $e$ the resulting distribution on $[0, 1]$ tells whether the first or the second coin in $e$ best fits the data. The idea is that if the data contains more heads than tails, then the first coin $e(0)$ is most likely: the learned distribution $r(0) + (1 - r)|1\rangle$ has $r > \frac{1}{2}$.

The second and third columns of the table below give the learned distributions on $[0, 1]$, both via C-learning and M-learning along the coin channel $e$. Each line describes a separate new learning action starting from the uniform distribution, independent from the outcomes of earlier lines. Explicitly, for point-data $\psi \in \mathcal{M}(\{H, T\})$ these updates of the uniform state $\nu \in \mathcal{D}([0, 1])$
These formulas give us the following table.

\[
\begin{align*}
\text{C-learning} & \quad \text{M-learning} \\
5|H\rangle + 5|T\rangle & = (4.19) 0.4491|0\rangle + 0.5509|1\rangle = 0.4949|0\rangle + 0.5051|1\rangle \\
9|H\rangle + 1|T\rangle & = 0.8050|0\rangle + 0.1951|1\rangle = 0.5354|0\rangle + 0.4646|1\rangle \\
8|H\rangle + 2|T\rangle & = 0.7335|0\rangle + 0.2665|1\rangle = 0.5253|0\rangle + 0.4747|1\rangle \\
4|H\rangle + 6|T\rangle & = 0.3522|0\rangle + 0.6478|1\rangle = 0.4848|0\rangle + 0.5152|1\rangle \\
7|H\rangle + 3|T\rangle & = 0.6472|0\rangle + 0.3528|1\rangle = 0.5152|0\rangle + 0.4848|1\rangle
\end{align*}
\]

These formulas give us the following table.

<table>
<thead>
<tr>
<th>point-data ψ</th>
<th>C-learning</th>
<th>M-learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>H⟩ + 5</td>
<td>T⟩</td>
</tr>
<tr>
<td>9</td>
<td>H⟩ + 1</td>
<td>T⟩</td>
</tr>
<tr>
<td>8</td>
<td>H⟩ + 2</td>
<td>T⟩</td>
</tr>
<tr>
<td>4</td>
<td>H⟩ + 6</td>
<td>T⟩</td>
</tr>
<tr>
<td>7</td>
<td>H⟩ + 3</td>
<td>T⟩</td>
</tr>
</tbody>
</table>

C-learning seems to perform binary classification best: picking up the differences between the numbers of heads and tails in the data. This C-learning column is as reported in [27], but uses higher precision. The term C-learning or the copied-state perspective do not occur in [27].

We conclude this section with some general observations. The most interesting point is that C-learning along a channel is additively compositional. This means that it can handle additional data as it arrives, by performing another C-learning step. This is a clear advantage of C-learning over M-learning.

**Proposition 4.8.7.** 1 Data transformation \( \omega_{\text{M}} \) from (4.19) is functorial in the sense that:

\[
\text{id} \approx_{\text{M}} \psi = \psi \quad \text{and} \quad (d \circ e) \approx_{\text{M}} \psi = e \approx_{\text{M}} (d \approx_{\text{M}} \psi).
\]

2 M-learning interacts well with channel composition:

\[
\text{Mlrn}(\omega, \text{id}, \psi) = \text{Mlrn}(\omega, \psi)
\]

\[
\text{Mlrn}(\omega, d \circ e, \psi) = \text{Mlrn}(\omega, e, d \approx_{\text{M}} \psi).
\]

3 C-learning is additively compositional, in the sense that:

\[
\text{Clrn}(\omega, e, \phi + \psi) = \text{Clrn}(\text{Clrn}(\omega, e, \phi), e, \psi).
\]

**Proof.** 1 We recall from Lemma 2.4.2 (5) and (6) that observable transformation is functorial, so that:

\[
id \approx (-) = \text{id} \quad \text{and} \quad (d \circ e) \approx (-) = (e \approx (-)) \circ (d \approx (-)).
\]
Chapter 4. Learning of States and Channels

The result then follows by functoriality of $M$.

2 By the previous point:

$$M \text{lrn}(\omega, \text{id}, \psi) = M \text{lrn}(\omega, \text{id} \ll_M \psi) = M \text{lrn}(\omega, \psi).$$

And:

$$M \text{lrn}(\omega, d \circ e, \psi) = M \text{lrn}(\omega, (d \circ e) \ll_M \psi) = M \text{lrn}(\omega, e \ll_M (d \ll_M \psi)) = M \text{lrn}(\omega, e, d \ll_M \psi).$$

3 Let’s use the ad hoc notation:

$$e \ll \psi := \&_p (e \ll p)^\psi(p).$$

We then have $e \ll (\phi + \psi) = (e \ll \phi) \& (e \ll \psi)$ since:

$$e \ll (\phi + \psi) = \&_p (e \ll p)^{(\phi + \psi)(p)}$$
$$= \&_p (e \ll p)^{\phi(p)} \& (e \ll p)^{\psi(p)}$$
$$= \&_p (e \ll p)^{\phi(p)} \& \&_p (e \ll p)^{\psi(p)}$$
$$= (e \ll \phi)(x) \& (e \ll \psi)(x).$$

Now we are almost done:

$$\text{Clrn}(\omega, e, \phi + \psi)$$
$$= \omega_{e \ll (\phi + \psi)}$$
$$= \omega_{e \ll \phi} \& (e \ll \psi)$$
$$= \text{Clrn}(\omega, e, \phi) \ll \psi$$
$$= \text{Clrn}(\text{Clrn}(\omega, e, \phi), e, \psi).$$

Exercises

4.8.1 Consider the channel $e: \{v, w\} \to \{a, b, c\}$ given by:

$$e(v) = \frac{1}{3}|a\rangle + \frac{2}{3}|b\rangle \quad \text{and} \quad e(w) = \frac{1}{2}|a\rangle + \frac{1}{2}|b\rangle + \frac{1}{3}|c\rangle.$$ 

with (uniform) state $\omega = \frac{1}{4}|u\rangle + \frac{1}{4}|v\rangle$ and point-data $\psi = 1|a\rangle + 2|b\rangle + 1|c\rangle$.

1 Show that:

$$e \ll_M \psi = 1\left|\frac{1}{3} \cdot 1_v + \frac{1}{3} \cdot 1_w\right\rangle + 2\left|\frac{2}{3} \cdot 1_v + \frac{1}{3} \cdot 1_w\right\rangle + 1\left|\frac{1}{3} \cdot 1_w\right\rangle.$$ 

2 Check that:

$$\omega \models e \ll_M \psi = \frac{539}{36584} \approx 0.0146$$

$$\omega \models e \ll_M \psi = \frac{871}{372} \approx 0.00586$$
3 Prove that:
\[
\text{Mlrn}(\omega, e, \psi) = \frac{36}{77}|v\rangle + \frac{41}{77}|w\rangle
\]
\[
\text{Mlrn}(\omega, e, \psi) \ll e \ll_M \psi = \frac{133476771}{8999178496} \approx 0.0148.
\]

4 Prove similarly that:
\[
\text{Clrn}(\omega, e, \psi) = |w\rangle
\]
\[
\text{Clrn}(\omega, e, \psi) \ll e \ll_M \psi = \frac{3}{556} \approx 0.0117.
\]

Given an informal explanation for why \(\text{Clrn}(\omega, e, \psi) = |w\rangle\).

4.8.2 In the setting of the previous exercise, consider the point-data \(\psi = |a\rangle + 2|b\rangle + |c\rangle\) from a missing-data perspective like in Subsection 4.2.1:

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>a</td>
</tr>
<tr>
<td>?</td>
<td>b</td>
</tr>
<tr>
<td>?</td>
<td>b</td>
</tr>
<tr>
<td>?</td>
<td>c</td>
</tr>
</tbody>
</table>

One can learn a distribution on \(X\), by updating the following joint state, for each of the four entries in this table:
\[
\tau := \langle \text{id}, e \rangle \gg v = \frac{1}{5}|v, a\rangle + \frac{4}{5}|v, b\rangle + \frac{1}{10}|w, a\rangle + \frac{4}{10}|w, b\rangle + \frac{3}{10}|w, c\rangle.
\]

Show that following the methodology of Subsection 4.2.2 for obtaining a distribution on \(X\) via updates of \(\tau\) gives the M-learned state.

4.8.3 In the setting of Example 4.8.4, consider the function:
\[
f(r) := (e(r) \ll_l \psi), \quad \text{for } r \in [0, 1].
\]

Prove that \(f\) takes its maximum at:
\[
15 + \sqrt{53, 809} = 0.6268214978709824
\]

\textit{Hint:} Compute the derivative of \(\ln \circ f\) and solve a second order equation.

4.8.4 Prove that for point-data \(\phi, \psi\),
\[
\text{Mlrn}(\omega \otimes \rho, c \otimes d, \phi \otimes \psi) = \text{Mlrn}(\omega, c, \phi) \otimes \text{Mlrn}(\rho, d, \psi).
\]
4.8.5 Prove that:

\[ e \gg \omega \bowtie \psi = \omega \bowtie e \bowtie_{M} \psi, \]

but in general:

\[ e \gg \omega \bowtie \psi \neq \omega \bowtie e \bowtie_{M} \psi \]

Hint: Recall from Exercise 2.4.1 that observable transformation \( \bowtie \) does not preserve conjunctions \&.

This last inequality \( \neq \) tells us that there is a second way to do C-learning along a channel, namely by increasing the validity \( e \gg \omega \bowtie \psi \) via updating \( \omega \) to \( \omega_{\nu_{\bowtie \psi}(x, p \bowtie \psi)} \). This however does not seem to make so much sense, especially when \( \psi \) consists of point-data, since \( 1_{x}^{\psi} = 1_{x} \).

4.9 Expectation Maximisation

Expectation-Maximisation (EM) is the name of a collection of learning techniques that typically use the sum-increase approach of Lemma 4.4.2 in a (joint) iteration of two separate steps, often referred to as ‘Expectation’ (E) and ‘Maximisation’ (M). For instance, Baum-Welch for learning new hidden Markov models (including Markov chains) in Sections 4.5 and 4.6, M-learning like in Example 4.8.3 and C-learning like in Example 4.8.6 are all called instances of Expectation-Maximisation. The first systematic exposition is in [21], but since then EM has developed into an umbrella term, see [74] for background information.

In the current channel-based setting we develop our own interpretation and elaboration of Expectation-Maximisation, where the previous section prepared the stage and developed the ‘E’ part, corresponding to learning a state along a channel. The ‘M’ part then consists of learning the channel as well. Hence for EM we can use the following picture, as refinement of (4.15):

\[ \text{channel to be learned} \]

\[ X \rightarrow Y \]

\[ \text{state to be learned} \]

\[ \text{data} \]

(4.23)

This picture indicates that EM is (potentially) a powerful technique: given two sets \( X, Y \) with data on \( Y \), it leads to a ‘latent’ state on \( X \) and to a channel \( X \rightarrow Y \) that fit the data. However, learning both is rather fragile and may lead to wildly different outcomes — in contrast to the situation in Section 4.8 where we were only learning the state and got good results. Indeed, we shall see that some —
4.9. Expectation Maximisation

naive but natural — approaches do not deliver as expected, for instance because they stabilise immediately, already after one iteration.

We continue the previous distinction between M-validity and C-validity to describe two possibilities for what ‘fit’ means, leading to two forms of Expectation-Maximisation, namely MEM and CEM. We also continue our view on data as consisting of a multiset of factors.

4.9.1 Multiple-state Expectation-Maximisation (MEM)

We first continue an earlier example of M-learning along a channel, in order to get a concrete idea of the issue at hand.

Example 4.9.1. In Example 4.8.3 with candies in a bag (from [83]), we started from a channel \( \langle f, w, h \rangle : \{0, 1\} \to \{C, L\} \times \{R, G\} \times \{H, H^*\} \) with data \( \psi \) on its codomain. This is used in what is now called the ‘E’ part of ‘EM’ to turn a prior state \( \rho \in \mathcal{D}(0, 1) \) into a new state \( \rho' := d \Rightarrow \mathrm{Firm}(\psi) \), where \( d = \langle f, w, h' \rangle : \{C, L\} \times \{R, G\} \times \{H, H^*\} \to \{0, 1\} \) is the Bayesian inversion (dagger).

As part of Maximisation in EM, we now also wish to update the three channels \( f : \{0, 1\} \to \{C, L\}, w : \{0, 1\} \to \{R, G\} \) and \( h : \{0, 1\} \to \{H, H^*\} \) via a ‘double dagger’. We take:

\[
dd' \coloneqq d'_{\mathrm{Firm}(\psi)} : \{0, 1\} \to \{C, L\} \times \{R, G\} \times \{H, H^*\}.
\]

We then obtain the individually learned channels \( f', w', h' \) via marginalisation of the channels:

\[
\begin{align*}
    f' & \coloneqq dd'[1, 0, 0] : \{0, 1\} \to \{C, L\} \\
    w' & \coloneqq dd'[0, 1, 0] : \{0, 1\} \to \{R, G\} \\
    h' & \coloneqq dd'[0, 0, 1] : \{0, 1\} \to \{H, H^*\}
\end{align*}
\]

This yields precisely the values reported in [83]:

\[
\begin{align*}
    f'(0) & = 0.6684|C\rangle + 0.3316|L\rangle & f'(1) & = 0.3887|C\rangle + 0.6113|L\rangle \\
    w'(0) & = 0.6483|R\rangle + 0.3517|G\rangle & w'(1) & = 0.3817|R\rangle + 0.6183|G\rangle \\
    h'(0) & = 0.6558|H\rangle + 0.3442|H^*\rangle & h'(1) & = 0.3827|H\rangle + 0.6173|H^*\rangle.
\end{align*}
\]

We generalise this approach as follows.

Theorem 4.9.2. Let \( e : X \to Y \) be a channel between finite sets \( X, Y \), with a state \( \omega \in \mathcal{D}(X) \) on its domain and with data \( \psi \in \mathcal{N}(\text{Fact}(Y)) \) on its codomain. We already know from [4.17] that \( \omega' = \text{Mlrn}(\omega, e, \psi) = \sum_{d \in \mathcal{D}(X)} \omega_{d,e} \psi_{d} \), giving a new state with increased M-validity: \( \omega' \models_{\mathcal{E}} e \preceq_{M} \psi \geq \omega \models_{\mathcal{E}} e \preceq_{M} \psi \).
1 We get a further increase \( \omega' \ll \psi \Rightarrow \omega' \ll \psi \) for the ‘better’ channel \( e' \): \( X \to Y \) defined by:

\[
e'(x) := \sum_p \frac{\psi(p)}{|\psi|} \cdot \frac{\omega_{|e'p}(x)}{\omega'(x)} \cdot e_p(x). \tag{4.24}
\]

2 In the special case when \( \psi \) consists of point-data, this formula gives the double-dagger:

\[
e' = (e'_\omega)_{\text{Flm}(\psi)} : X \to Y. \tag{4.25}
\]

**Proof.** The proof proceeds as for Proposition 4.4.6. The result follows from the inequality, for factors \( p_1, \ldots, p_n \),

\[
\prod_i (\omega \mid e \ll p_i) \leq \prod_i (\omega \mid e' \ll p_i)
\]

for the channel \( e' \) given by the following convex combination of updated channels:

\[
e'(x)(y) = \sum_i \frac{\omega_{|e'p_i}(x)}{\sum_j \omega_{|e'p_j}(x)} \cdot e_{p_i}(x)(y).
\]

We shall derive this outcome for only two predicates \( p_1, p_2 \) and leave the \( n \)-ary generalisation to the interested reader.

The proof uses Lemma 4.4.2 (again), now with function

\[
F(e, x, x', y, y') := \omega(x) \cdot e(x)(y) \cdot p_1(y) \cdot \omega(x') \cdot e(x')(y') \cdot p_2(y').
\]

Then: \( \sum_{x, x', y, y'} F(e, x, x', y, y') = (\omega \mid e \ll p_1) \cdot (\omega \mid e \ll p_2) \).

We write \( X = \{x_1, \ldots, x_n\} \) and \( Y = \{y_1, \ldots, y_m\} \) and use variables \( z_{ij} \) for the channel \( e' \) that we are seeking. We do so via the auxiliary function:

\[
H(\vec{z}, \vec{\lambda}) := \sum_{i,j,f} F(e, x_i, x_f, y_j, y_f) \cdot \ln \left( \omega(x_i) \cdot z_{ij} \cdot p_1(y_j) \cdot \omega(x_f) \cdot z_{jf} \cdot p_2(y_f) \right) - \sum_i \lambda_i \cdot ((\sum_{j} z_{ij}) - 1).
\]

We then have partial derivatives:

\[
\frac{\partial H}{\partial z_{kl}}(\vec{z}, \vec{\lambda}) = \sum_{i,j,f} F(e, x_k, x_f, y_k, y_f) \cdot \frac{1}{z_{kl}} + \sum_i F(e, x_k, x_l, y_f) \cdot \frac{1}{z_{kl}} - \lambda_k
\]

\[
= \frac{\omega(x_k) \cdot e(x_k)(y_f) \cdot p_1(y_f) \cdot (\omega \mid e \ll p_2)}{z_{kl}}
\]

\[
+ \frac{\omega(x_k) \cdot e(x_k)(y_f) \cdot p_2(y_f) \cdot (\omega \mid e \ll p_1)}{z_{kl}} - \lambda_k
\]

\[
\frac{\partial H}{\partial \lambda_k}(\vec{z}, \vec{\lambda}) = (\sum_j z_{kj}) - 1.
\]
We set all these partial derivatives to zero and deduce that for each $k$ the following fraction equals one:
$$
\omega(x_k) \cdot (e \ll p_1)(x_k) \cdot (\omega \models e \ll p_2) + \omega(x_k) \cdot (e \ll p_2)(x_k) \cdot (\omega \models e \ll p_1).
$$

Hence we get as better, newly learned channel:
$$
e'(x_k)(y_\ell) = z_{k\ell}
= \frac{\omega(x_k) \cdot e(x_k)(y_\ell) \cdot p_1(y_\ell) \cdot (\omega \models e \ll p_2)}{p_1(x_k)(y_\ell)} + \frac{\omega(x_k) \cdot e(x_k)(y_\ell) \cdot p_2(y_\ell) \cdot (\omega \models e \ll p_1)}{p_1(x_k)(y_\ell)}
= \frac{\omega(x_k) \cdot (e \ll p_1)(x_k) \cdot (\omega \models e \ll p_2)}{p_1(x_k)} \cdot e_{p_1}(x_k)(y_\ell)
+ \frac{\omega(x_k) \cdot (e \ll p_2)(x_k) \cdot (\omega \models e \ll p_1)}{p_1(x_k)} \cdot e_{p_1}(x_k)(y_\ell)
= \frac{\omega_{e \ll p_1}(x_k)}{\omega_{e \ll p_1}(x_k) + \omega_{e \ll p_2}(x_k)} \cdot e_{p_1}(x_k)(y_\ell)
+ \frac{\omega_{e \ll p_2}(x_k)}{\omega_{e \ll p_1}(x_k) + \omega_{e \ll p_2}(x_k)} \cdot e_{p_2}(x_k)(y_\ell).
$$

We now turn to the special case where we have point-data $\psi \in M(\mathcal{Y})$ and show how to derive \[4.25\] from \[3.24\]. Recall that Lemma \[4.8.2\] tells that in this situation $\omega' = e_\omega \gg \text{Flrn}(\psi)$. Since $e_{1\psi}(x) = e(x)|_{1\psi} = 1|\psi$ we get:
$$
e'(x) \sum_{\psi} \frac{\psi(y)}{|\psi|} \cdot \frac{\omega_{e \ll p_1}(x)}{\omega'(x)} \cdot e_{1\psi}(x)
= \sum_{\psi} \frac{\text{Flrn}(\psi)(y) \cdot e_\omega^\dagger(y)}{(e_\omega \gg \text{Flrn}(\psi))(x)} \cdot 1|\psi
= \sum_{\psi} \frac{(e_\omega)^\dagger_{\text{Flrn}(\psi)}(x)(y)}{|\psi|} \cdot 1|\psi
= (e_\omega)^\dagger_{\text{Flrn}(\psi)}(x).$$

Point \[2\] of this result covers the double-dagger that we have seen in Example \[4.9.1\] with the candies from \[88\]. It is worth observing that, in the situation of point \[2\] we get maximal M-validity, since for point-data $\psi$ we have:
$$
e' \gg \omega' \models \psi = (e_\omega)^\dagger_{\text{Flrn}(\psi)} \gg (e_\omega \gg \text{Flrn}(\psi)) \models \psi \quad \text{by Lemma \[4.8.2\]}
= \text{Flrn}(\psi) \models \psi \quad \text{by \[3.22\].}$$

263
The latter M-validity is maximal, by Proposition 4.7.5.

In fact, repeating M-learning along a channel, with point-data, makes no sense, because the process reaches a fixed point in one step already. That is the content of the following result.

**Proposition 4.9.3.** Consider a channel \( c : X \rightarrow Y \) with states \( \sigma \in \mathcal{D}(X) \) and \( \tau \in \mathcal{D}(Y) \). Define:

\[
d := c^\dagger \vdash : Y \rightarrow X \quad \sigma' := d \gg \tau \in \mathcal{D}(X).
\]

If we take the double-dagger of \( c \), like in Theorem 4.9.2(2):

\[
c' := d' \vdash : X \rightarrow Y,
\]

then its dagger satisfies: \( c'^\dagger = d = c^\dagger \). We then also get \( c'^\dagger \gg \tau = d \gg \tau = \sigma' \).

**Proof.** We give both a concrete and an abstract proof. For the concrete proof we first note:

\[
c' \gg \sigma' = d' \gg (d \gg \tau) \approx \tau.
\]

But then:

\[
c'^\dagger \psi \approx (d \gg \tau) \psi = d \psi.
\]

The more abstract proof uses the category \( \text{Krn} \) of kernels from Definition 3.8.2. It allows us to write the channel \( d \) as a morphism in \( \text{Krn} \) of the form:

\[
(Y, \tau) \xrightarrow{d} (X, \sigma').
\]

Theorem 3.8.3 says that taking the dagger of a morphism twice returns the original morphism. In this case it gives \( c' = d'' = d \).

This result puts cold water on MEM and severely limits its usability. Recall that in Section 4.8 we did get progress via multiple iterations of M-learning — notably in Examples 4.8.3, 4.8.4 and 4.8.5 — but there we kept the channel \( e \) fixed and iterated the learning of new states, along the fixed channel.

**Remark 4.9.4.** 1 It is an intriguing question whether the authors of [88] are aware that the form EM that they use (MEM) stabilises after one iteration. It seems they are not: they calculate log-likelihoods, amounting to:

\[
\ln \left( \langle f, w, h \rangle \gg \rho \| \psi \right) \approx -2044,
\]

and also:

\[
\ln \left( \langle f', w', h' \rangle \gg \rho' \| \psi \right) = -2021.
\]
4.9. Expectation Maximisation

This gives the expected increase (as in Theorem 4.9.2). But then we find in [88] §20.3:

By the tenth iteration, the learned model is a better fit than the original model ($L = -1982.214$).

This is a mysterious remark. First of all, if you know that things stabilise after one iteration, you are not going to iterate ten times. Secondly, it is unclear where the likelihood $L = -1982.214$ comes from. The closes thing that we can reconstruct is:

$$\ln \langle dd \gg \rho' \gg \psi \rangle \approx -1979,$$

where $dd = d'_{\text{Flrn}(\psi)}$ is the double-dagger introduced in Example 4.9.1.

Aside: we may indeed expect this value $-1979$ to be higher than the previous one $-2021$ since the latter involves a tuple of marginalisations $(f', w', h')$ taken from $dd$.

2 Topic modelling is an area where one uses statistical techniques to identify a (previously fixed) number of topics in documents based on word occurrences. One approach, called Probabilistic Latent Semantic Analysis [38], is an instance of the present MEM setting. We assume that we have finite sets $D$ of documents, $W$ of words, and $T$ of topics. A document-word matrix is given as $\psi \in N(D \times W)$, such that $\psi(d, w) \in \mathbb{N}$ is the number of occurrences of word $w$ in document $d$. We consider $\psi$ as point-data. The aim is to learn two channels $e: T \rightarrow D$ and $f: T \rightarrow W$ which assign to a topic $z \in T$ the distribution $e(z) \in \mathcal{D}(D)$ of documents, associated with topic $z$, and the distribution $f(z) \in \mathcal{D}(W)$ of words, associated with topic $z$. Thus $e(z)(d)$ is the probability that document $d$ is on topic $z$ and $f(z)(w)$ is the probability that word $w$ belongs to topic $z$.

As usual, a prior distribution $\sigma \in \mathcal{D}(T)$ of topics is assumed, possibly uniform initially. We plan to learn an updated distribution along the tuple channel $(e, f): T \rightarrow D \times W$, using point-data $\psi$. Following MEM we can update $\sigma$ to:

$$\sigma' := \text{Flrn}(\sigma, (e, f), \psi) \quad \langle e, f \rangle_{\sigma'} \Rightarrow \text{Flrn}(\psi).$$

The dagger channel $(e, f)^{\dagger}_{\sigma'}: D \times W \rightarrow T$ is described concretely in [38] Eqn. (6) as conditional probability $P(z \mid d, w)$, satisfying:

$$P(z \mid d, w) = \frac{P(z) \cdot P(d \mid z) \cdot P(w \mid z)}{\sum_{z'} P(z') \cdot P(d \mid z') \cdot P(w \mid z')}.$$

This corresponds precisely to the dagger that we use, since:

$$\langle e, f \rangle_{\sigma'} (d, w)(z) = \frac{\sigma(z) \cdot (e, f)(z)(d, w)}{(e, f) \gg \sigma')(d, w)} = \frac{\sigma(z) \cdot e(z)(d) \cdot f(z)(w)}{\sum_{z'} \sigma(z') \cdot e(z')(d) \cdot f(z')(w)}.$$
Newly learned channels $e': T \rightarrow D$ and $f': T \rightarrow W$ are obtained in our setting via Theorem 4.9.2 (2), namely by first taking the double dagger $g: T \rightarrow D \times W$, from which $e'$ and $f'$ are obtained by marginalisation:

$$g = ((e, f)^\dagger)^\dagger_{\text{Flrn}(\psi)}$$

with

$$\begin{cases} e' := g[1, 0] \\ f' := g[0, 1] \end{cases}$$

More concretely,

$$e'(z)(d) = \sum_w \text{Flrn}(\psi)(d, w) \cdot \langle e, f \rangle^\dagger_{\sigma(w)}(d, w)(z)$$

This is the newly learned conditional probability $P(d \mid z)$ described in [38, Eqn. (4)]. It is unclear to what extent it is recognized in [38] that this MEM approach stabilises after one round — as shown in Proposition 4.9.3.

### 4.9.2 Copied-state Expectation-Maximisation (CEM)

For the next form of Expectation-Maximisation we also first continue an earlier illustration.

**Example 4.9.5.** Recall Example 4.8.6 on classification of coins, where we have a channel $e: \{0, 1\} \rightarrow \{H, T\}$ and a uniform prior state $\nu$ on $\{0, 1\}$. In this situation we have five pieces of point-data $\psi_1, \ldots, \psi_5 \in N([H, T])$, giving rise to five newly learned states on $\{0, 1\}$ via C-learning along $e$, namely:

$$\omega_i := \text{Clrn}(\nu, e, \psi_i) = \nu_{\langle e, \psi_i \rangle}$$

see Table (4.22) for details about this expectation-part of EM.

The maximisation-part of EM in [27] can be described as follows. Five product states $\omega_i \otimes \text{Flrn}(\psi_i) \in D(\{0, 1\} \times [H, T])$ are formed and added up in a (uniform) convex sum:

$$\tau := \sum_i \frac{1}{5} \cdot (\omega_i \otimes \text{Flrn}(\psi_i))$$

A new channel $e' = \tau[0, 1 \mid 1, 0]: \{0, 1\} \rightarrow [H, T]$ is then extracted, which can be described more concretely as follows (with the same parameters as in [27]):

$$e'(0) = 0.713[H] + 0.287[T] \quad e'(1) = 0.5813[H] + 0.4187[T].$$

As newly learned state we can take $\omega' = \tau[1, 0] = \sum_i \frac{1}{5} \cdot \omega_i$. What has happened here?

An aside: in the light of Remark 4.1.3 we notice that we do not need to force
4.9. Expectation Maximisation

\[ \tau \] to be state, if we use it only for disintegration. We might as well define it as a multiset, of the form \( \tau = \sum \omega_i \otimes \psi_i \), where we view the states \( \omega_i \) as multisets. We can then still define \( e' = \tau[0,1|1,0] \). This makes the calculations a bit simpler.

In the remainder of this section we analyse the above situation. We do not do this in full generality, but only for point-data, since it is unclear how to handle the general case — with multisets of factors as data.

We start with a negative, but revealing observation, telling that the sum-increase approach that we have used so succesfully so far, see Lemma 4.4.2, is not very helpful here.

**Lemma 4.9.6.** Assume we have a channel \( e : X \to Y \) with a state \( \omega \in \mathcal{D}(X) \) and point-data \( \psi \in \mathcal{M}(Y) \). Finding a better channel \( e' : X \to Y \) for C-learning along \( e \), via the sum-increase method, yields the constant channel \( e'(x) = \text{Flrn}(\psi) \).

**Proof.** We are interested in increasing the C-validity:

\[ \omega \models e \ll_M \psi = \omega \models &_{\cdot}(e \ll 1_y)^{\psi(y)} = \sum e(\omega) \cdot \prod e(\psi(y))^{\psi(y)}. \]

Our aim is to do so via a better channel \( e' \). Therefor we look at the function:

\[ F(e, x) := \omega(x) \cdot \prod e(\psi(y))^{\psi(y)} = \prod \omega(x) \cdot e(\psi(y))^{\psi(y)}. \]

We write \( X = \{x_1, \ldots, x_n\} \) and \( Y = \{y_1, \ldots, y_m\} \) and look for \( z_{ij} \) representing \( e'(x_i)(y_j) \). We use the function:

\[ H(\hat{z}, \hat{\lambda}) := \sum e(x, i) \cdot \ln(\prod \omega(x_i) \cdot z_{ij}) - \sum \lambda_i \cdot ((\sum z_{ij}) - 1). \]

It has partial derivatives:

\[
\frac{\partial H}{\partial z_{ij}}(\hat{z}, \hat{\lambda}) = F(e, x_i) \cdot \psi(y_j), \quad \frac{\partial H}{\partial \lambda_k}(\hat{z}, \hat{\lambda}) = (\sum z_{ij}) - 1.
\]

When we set these to zero we get:

\[ 1 = \sum \frac{F(e, x_i) \cdot \psi(y_j)}{\lambda_k}, \quad \frac{F(e, x_i) \cdot \psi(y_j)}{\lambda_k} = F(e, x_i) \cdot \psi(y_j) = \text{Flrn}(\psi)(y_j). \]

Hence:

\[ z_{ij} = \frac{F(e, x_i) \cdot \psi(y_j)}{\lambda_k} = \frac{F(e, x_i) \cdot \psi(y_j)}{F(e, x_i) \cdot \psi(1)} = \frac{\psi(y_j)}{\psi} = \text{Flrn}(\psi)(y_j). \]

Getting a constant channel is not very useful in an iterative learning process. But that is not what is used in Example 4.9.5. The crucial difference there is that we have multiple point-data elements \( \psi_i \) from which we learn a single
channel. It turns out that what happens there is a mixture of C-learning — to get multiple states \( \omega_i \), one for each \( \psi_i \) — and M-learning — to get the new channel \( e' \). This is what we will look at next.

### 4.9.3 Combined M- and C-versions of Expectation-Maximisation (MCEM)

Starting in Section 4.7 we have described data as a multiset of factors, with a multiset of elements as a special case — called point-data. But what if we have multiple portions of data, as in Example 4.9.5 where there are five multisets from which to learn. How do we handle such multiple portions more abstractly, and what is the appropriate form of validity? In this section we briefly look into this matter, without claiming to have the definitive story on this matter.

We have identified data on a set \( X \) with a multiset of factors \( \psi \in M(\text{Fact}(X)) \). But what if we have multiple data \( \psi_i \)? How to organise them? Well, we can simply climb the abstraction ladder a bit further, and use multisets of multisets of factors, of the form \( \Psi \in M(M(\text{Fact}(X))) \). As special case we can use point-predicates as factors, which we can identify with a multiset of multisets \( \Psi \in M(M(X)) \).

For instance, in Example 4.9.5 one has such a multiset of multisets, namely \( \Psi = 1|\psi_1 \rangle + 1|\psi_2 \rangle + 1|\psi_3 \rangle + 1|\psi_4 \rangle + 1|\psi_5 \rangle \).

Now let’s return to the general case of \( \Psi \in M(M(\text{Fact}(X))) \). What would be its validity be in a state \( \omega \). One possible, quite natural, interpretation is to use a combination of M- and C-validity, as in:

\[
\omega \models_{MC} \Psi := \prod_{\phi} \left( \omega \models_{C} \phi \right)^{\Psi(\phi)} \prod_{\phi} \left( \omega \models_{M} \phi^{p(\phi)} \right)^{\Psi(\phi)}.
\]

Here we use the M-validity-approach on the outside, with C-validity inside. There is no obvious way to it the other way around. This MC-validity includes both M-validity and C-validity as special cases, see Exercise 4.9.3.

The form of this validity definition suggests how to do MC-learning, namely as M-learning with the data \( \Psi \in M(\text{Fact}(X)) \) given by:

\[
\Psi := \sum_{\phi} \Psi(\phi) \quad \text{so that} \quad \omega \models_{MC} \Psi = \omega \models_{M} \Psi.
\]

Thus we can express MC-learning in terms of M-learning:

\[
\text{MClrn}(\omega, \Psi) := M\text{lrn}(\omega, \Psi)
\]

\[
= \sum_{\phi} \Psi(\phi) \quad \text{by Theorem 4.7.3}\]

\[
= \sum_{\phi} \frac{\Psi(\phi)}{|\Psi|} \text{Clrn}(\omega, \phi) \quad \text{by Theorem 4.7.2}.\]

The next question is: what if we have a channel \( e: X \Rightarrow Y \), with a state
We then get:

\[
e \ll_{\text{MC}} \Psi := \mathcal{M}(e \ll (-))(\Psi) = \sum_{\varphi} \Psi(\varphi) | M(e \ll (-))(\varphi) \rangle = \sum_{\varphi} \Psi(\varphi) e \ll_{\text{MC}} \varphi \rangle = \sum_{\varphi} \Psi(\varphi) \sum_p \varphi(p) | e \ll p \rangle.
\]

(4.28)

We then get:

\[
\omega \ll_{\text{MC}} e \ll_{\text{MC}} \Psi \overset{(4.29)}{=} \prod_{\varphi} (\omega \ll e \ll_{\text{MC}} \varphi)^{\Psi(\varphi)} = \prod_{\varphi} (\omega \ll \&_p (e \ll p)^{\varphi}(p)^{\Psi(\varphi)}).
\]

The transformation \( e \ll (-) \) now occurs within the conjunction \&. This is significant, since such factor transformations and conjunctions do not commute, see Exercise 2.4.1. As a result, we cannot reduce MC-learning along a channel. However, we can say something when we have a multiset of point-data.

**Theorem 4.9.7.** Let \( e : X \rightarrow Y \) be a channel with a state \( \omega \in \mathcal{D}(X) \). For a multiset \( \Psi \in \mathcal{N}(N(Y)) \) of point-data we are interested in increasing the "MC" validity:

\[
\omega \ll_{\text{MC}} e \ll_{\text{MC}} \Psi = \prod_{\varphi} (\omega \models e \ll_{\text{MC}} \varphi)^{\Psi(\varphi)}.
\]

Then we can get a better state and channel via MC-learning:

\[
\omega' := \sum_{\varphi} \Psi(\varphi) \cdot \text{Clrn}(\omega, e, \varphi)
\]

\[
e'(y) := \sum_{\varphi} \Psi(\varphi) \cdot \varphi(y) \cdot \text{Clrn}(\omega, e, \varphi)(y).
\]

(4.30)

**Proof.** We keep things simple and shall prove the result for the special case where \( \Psi = 1_{\varphi_1} + 1_{\varphi_2} \). We thus wish to increase the product:

\[
(\omega \models e \ll_{\text{MC}} \varphi_1) \cdot (\omega \models e \ll_{\text{MC}} \varphi_2),
\]

by finding both a better \( \omega \) and \( e \). We thus look at the function:

\[
F(\omega, e, x, x') := \omega(x) \cdot (\&_y (e \ll 1_y)^{\varphi(y)}(x) \cdot \omega(x') \cdot (\&_y (e \ll 1_y)^{\varphi(y)}(x'))
\]

\[
= \prod_{x,y} \omega(x) \cdot e(x)(y)^{\varphi(y)} \cdot \omega(x') \cdot e(x')(y')^{\varphi(y')}.
\]

As we have done before, we write \( X = \{x_1, \ldots, x_n\} \) and \( Y = \{y_1, \ldots, y_m\} \) and
use variables \( w_i \) for \( \omega'(x_i) \) and \( z_{ij} \) for \( \epsilon'(x_i,y_j) \). In order to use Lemma 4.4.2, we define:

\[
H(\vec{w}, z, \kappa, \vec{\lambda}) := \sum_{i} F(\omega, e, x_i, x_i') \cdot \ln \left( \prod_{j} w_j \cdot z_{ij} \cdot \epsilon_{\gamma}^{(i)} \cdot \omega_{\gamma}^{\epsilon_{\gamma}} \right)
- \kappa \cdot \left( \sum_i w_i - 1 \right) - \sum_i \lambda_i \cdot \left( \sum_j z_{ij} - 1 \right)
= \sum_{i,j} F(\omega, e, x_i, x_i') \cdot \left( \ln(w_i) + \ln(w_j') + \phi_1(y_j) \cdot \ln(z_{ij}) + \phi_2(y_j') \cdot \ln(z_{ij}') \right)
- \kappa \cdot \left( \sum_i w_i - 1 \right) - \sum_i \lambda_i \cdot \left( \sum_j z_{ij} - 1 \right).
\]

We first look at the following two partial derivatives.

\[
\frac{\partial H}{\partial \kappa}(\vec{w}, z, \kappa, \vec{\lambda}) = (\sum_i w_i) - 1
\]

\[
\frac{\partial H}{\partial w_k}(\vec{w}, z, \kappa, \vec{\lambda})
= \sum_{i,j} \frac{F(\omega, e, x_i, x_i')}{w_k} + \sum_i \frac{F(\omega, e, x_i, x_k)}{w_k} - \kappa
= \prod_j \omega(x_i) \cdot (e \ll 1_j)(x_i) \cdot (\omega \models \epsilon \ll 1_j e_{\gamma}^{\epsilon_{\gamma}})
+ \frac{(\omega \models \epsilon \ll 1_j e_{\gamma}^{\epsilon_{\gamma}}) \cdot \prod_j \omega(x_i) \cdot (e \ll 1_j)(x_i) e_{\gamma}^{\epsilon_{\gamma}}}{w_k} - \kappa
= \frac{\omega \models e \ll \Psi \models}{w_k} \cdot \left( Clr(\omega, e, \phi_1)(x_i) + Clr(\omega, e, \phi_2)(x_i) \right) - \kappa.
\]

Setting these to zero gives:

\[
1 = \sum_k w_k
= \sum_k \frac{\omega \models e \ll \Psi \models}{w_k} \cdot \left( Clr(\omega, e, \phi_1)(x_i) + Clr(\omega, e, \phi_2)(x_i) \right)
= \frac{2}{\kappa} \cdot \left( \omega \models e \ll \Psi \models \right).
\]

Hence \( \kappa = 2 \cdot (\omega \models e \ll \Psi) \), so that:

\[
\omega'(x_i) = w_k = \frac{1}{2} \cdot Clr(\omega, e, \phi_1)(x_i) + \frac{1}{2} \cdot Clr(\omega, e, \phi_2)(x_i).
\]

Thus, the newly learned state \( \omega' \) is a convex combination of C-learned states along \( e \).
4.9. Expectation Maximisation

We turn to the other two partial derivatives of $H$.

$$\frac{\partial H}{\partial \lambda_k}(\vec{w}, \vec{z}, \kappa, \vec{\lambda}) = (\sum_j z_{k,j}) - 1$$

$$\frac{\partial H}{\partial z_{k\ell}}(\vec{w}, \vec{z}, \kappa, \vec{\lambda}) = \sum_i \frac{F(\omega, e, x_i, x_i) \cdot \varphi_1(y_i)}{z_{k\ell}} + \sum_i \frac{F(\omega, e, x_i, x_i) \cdot \varphi_2(y_i)}{z_{k\ell}} - \lambda_k$$

$$= \prod_j \omega(x_j) \cdot (e \ll 1_j) \cdot (\omega \not= \&_y (e \ll 1_j)^{\varphi_2(y)}) \cdot \varphi_1(y_j)$$

$$\text{But then:}$$

$$e'(x_k)(y_i) = z_{k\ell} \cdot \frac{\varphi_1(y_i) \cdot Clrn(\omega, e, \varphi_1)(x_k) + \varphi_2(y_i) \cdot Clrn(\omega, e, \varphi_2)(x_k)}{\lambda_k}$$

When these partial derivatives are zero we can derive:

$$1 = \sum_j z_{k\ell}$$

$$= \sum_j \frac{\omega \not= e \ll \kappa \cdot \Psi}{\lambda_k} \cdot \left(\frac{\varphi_1(y_i) \cdot Clrn(\omega, e, \varphi_1)(x_k) + \varphi_2(y_i) \cdot Clrn(\omega, e, \varphi_2)(x_k)}{\lambda_k}\right)$$

$$= \frac{\omega \not= e \ll \kappa \cdot \Psi}{\lambda_k} \cdot \left(\frac{[\varphi_1] \cdot Clrn(\omega, e, \varphi_1)(x_k) + [\varphi_2] \cdot Clrn(\omega, e, \varphi_2)(x_k)}{\lambda_k}\right).$$

But then:

$$e'(x_k)(y_i) = z_{k\ell} \cdot \frac{\varphi_1(y_i) \cdot Clrn(\omega, e, \varphi_1)(x_k) + \varphi_2(y_i) \cdot Clrn(\omega, e, \varphi_2)(x_k)}{\lambda_k}. \quad \Box$$

What remains is to show that the $\omega'$ and $e'$ in Example [4.9.5] are instances of the formulas (4.30) for $\Psi = \sum_{i \leq 5} 1|\psi_i$. This is immediate for:

$$\omega' = \sum_j \frac{1}{2} \cdot \omega_j = \sum_i \frac{\Psi(\psi_i)}{|\Psi|} \cdot Clrn(\omega, e, \psi_i).$$

For $e' = \tau|0, 1 \mid 1, 0$ where $\tau = \sum \omega_i \otimes \psi_i$ we have:

$$e'(x)(y) := \frac{\tau(x, y)}{\sum \tau(x, z)} = \frac{\sum_i \omega(x_i) \cdot \psi_i(y)}{\sum_i \omega(x_i) \cdot \psi_i(z)} = \frac{\sum_i \Psi(\psi_i) \cdot Clrn(\omega, e, \psi_i)(x) \cdot \psi_i(y)}{\sum_i \Psi(\psi_i) \cdot Clrn(\omega, e, \psi_i)(x) \cdot |\psi|}. $$

271
We conclude that what is called Expectation-Maximisation in [27] about Example 4.9.5 is a mixture of C-learning and M-learning.

When this form of learning is repeated 10 times, each time with the same data, one gets the following state and channel.

$$\omega = 0.5376|0\rangle + 0.4624|1\rangle$$

$$\left\{ \begin{array}{l}
e(0) = 0.7899|H\rangle + 0.2101|T\rangle \\
ne(1) = 0.5089|H\rangle + 0.4911|T\rangle.
\end{array} \right.$$

These outcomes are basically the same as in [27], where the small differences seem to be due to rounding errors: [27] uses only two decimals, probably also during the computations.

### Exercises

**4.9.1** Consider point-data

$$\psi = 3|0\rangle + 2|1\rangle + 5|2\rangle \in \mathcal{M}(3),$$

together with the parameterised state

$$\text{flip}(r) = r|1\rangle + (1-r)|0\rangle \in \mathcal{D}(2)$$

and the parameterised channel

$$c(s,t): 2 \rightarrow 3$$

given by:

$$c(s,t)(0) = s|0\rangle + (1-s)|1\rangle$$

$$c(s,t)(1) = (1-t)|1\rangle + t|2\rangle.$$

The parameters $$r, s, t$$ are all in the unit interval $$[0, 1].$$

1. Compute $$c(s,t) \gg \text{flip}(r).$$
2. Show that the equation $$c(s,t) \gg \text{flip}(r) = \text{Flrn}(\psi)$$ yields the equations:

$$s = \frac{3}{10r}, \quad t = \frac{1}{2(1-r)}$$

for $$r \in (0, 1).$$ This gives a family of solutions.

3. Now let’s take $$r = s = t = \frac{1}{2}$$ and write $$\omega = \text{flip}(\frac{1}{2})$$ and $$e = c(\frac{1}{2}, \frac{1}{2}).$$

Our aim is to learn better values for $$r, s, t$$ via the M-learning version of EM, starting with $$\omega, e.$$ Compute $$e^{\dagger}_{\omega}: 3 \rightarrow 2$$ and show that:

$$Mlrn(\omega, e, \psi) \overset{(4.19)}{\Rightarrow} e^{\dagger}_{\omega} \gg \text{Flrn}(\psi) = \frac{3}{2}|0\rangle + \frac{3}{2}|1\rangle.$$  

4. Show that the double-dagger $$(e^{\dagger}_{\omega})_{\text{flip}(\psi)}: 2 \rightarrow 3$$ satisfies:

$$(e^{\dagger}_{\omega})_{\text{flip}(\psi)}(0) = \frac{3}{4}|0\rangle + \frac{1}{4}|1\rangle, \quad (e^{\dagger}_{\omega})_{\text{flip}(\psi)}(1) = \frac{1}{8}|1\rangle + \frac{5}{8}|2\rangle.$$  

5. Check that the values $$r = \frac{3}{4}$$ and $$s = \frac{3}{4}, t = \frac{5}{8}$$ that emerge from MEM in points (3) and (4) are instances of the family of solutions described in (2).
4.9.2 Consider the topic modelling situation from Remark 4.9.4. There we have a joint distribution \( \tau \in D(D \times W) \) on documents of words given by:

\[
\tau = \langle e, f \rangle \gg \sigma.
\]

Show that the channel \( r[0, 1] \mid 1, 0] \colon D \to W \), extracted via disintegration, equals \( f \circ e^\dagger \), see also [38, Eqn. (7)].

4.9.3 Let \( \Phi \in M(\text{Fact}(X)) \) be data on a set \( X \). There are two ways of turning \( \Phi \) into a multiset in \( M(M(\text{Fact}(X))) \), via the unit maps \( \text{unit} : A \to M(A) \), given by \( \text{unit}(a) = 1|a \rangle \) from Subsection 4.4.2 Concretely:

\[
\text{unit}(\Phi) = 1|\Phi \rangle \quad M(\text{unit})(\Phi) = \sum_p \Phi(p)|\text{unit}(p)\rangle = \sum_p \Phi(p)|1|p\rangle\rangle.
\]

Check that:

\[
\omega \mid_\Phi \text{unit}(\Phi) = \omega \mid_\Phi \Phi \quad \omega \mid_\Phi M(\text{unit})(\Phi) = \omega \mid_\Phi \Phi.
\]

4.9.4 We now use the approach of the previous exercise in the setting of Theorem 4.9.7. Let \( \omega \in D(X) \), \( e : X \to Y \) and \( \Phi \in M(Y) \).

1 Show that the newly learned state and channel \( \omega' \) and \( e' \) in (4.30) from \( \text{unit}(\Phi) \in M(M(Y)) \) are:

\[
\omega' = \text{Clrn}(\omega, e, \Phi) \quad e'(x) = \text{Flrn}(\Phi).
\]

2 Show also that \( \omega' \) and \( e' \) obtained from \( M(\text{unit})(\Phi) \in M(M(Y)) \) are:

\[
\omega' = M\text{lrn}(\omega, e, \Phi) \quad e' = (e^\dagger \omega)^\dagger_{\text{Flrn}(\Phi)}.
\]

4.10 A clustering example

This section describes some experiments with Expectation-Maximisation for classifying clustered points in a square grid. The grid that we use is discrete and has size 30 \( \times \) 30. We randomly choose three centers in the plane and then randomly choose 10 points around these centers (which may possibly overlap). This gives pictures as in the column on the left in Figure 4.3 where chosen points are indicated by crosses \( \times \).

Our aim is to classify the given points in the plane into one of three categories, indicated by colours yellow (Y), red (R) or blue (B). Several example classifications are in the column on the right in Figure 4.3 Visually it is clear
which classifications succeed (partially) or fail. We construct such a classification algorithm based on the techniques from the previous sections. There may be other techniques, which perform far better, but we will use what we have seen so far, as illustration.

This situation is an instance of the Expectation-Maximisation set-up in Diagram (4.23), with a channel of the form:

\[ (Y, R, B) = X \rightarrow Y = 30 \times 30 \]

The crossed points form the data on the codomain \( Y = 30 \times 30 \). Our aim is to learn both a state \( \omega \) on \( X = \{Y, R, B\} \) and a channel \( e : X \rightarrow Y \). For each crossed point \( y = (y_1, y_2) \in 30 \times 30 \) we can then compute its classification in \( \{Y, R, B\} \) via the dagger of the channel \( e \), namely via the distribution:

\[ e_\omega^\dagger (y) \in D((Y, R, B)). \]

The colour that we assign to point \( y \) is the argmax of this distribution: the colour with the highest probability, see Subsection 1.4.4. This is an important methodological point: we classify via the dagger of the channel in an EM learning situation (4.23).

In order to learn the state \( \omega \) and channel \( e \) we massage the data into the right form. Each crossed point \( y = (y_1, y_2) \in 30 \times 30 \) is turned into a predicate \( p_y : 30 \times 30 \rightarrow \{0, 1\} \). It is zero almost everywhere on \( 30 \times 30 \), except on 9 places (1% only), as indicated by the sub-grid of probabilities below, with crossed point \( y \) in the center.

\[
\begin{array}{ccc}
\frac{1}{2} & \frac{3}{4} & \frac{1}{2} \\
\mid & | & | \\
\frac{3}{4} & 1 & \frac{3}{4} \\
\mid & | & | \\
\frac{1}{2} & \frac{3}{4} & \frac{1}{2} \\
\end{array}
\]

These crossed points are chosen to be never on the border of the grid, so there is enough room around them to define a predicate in this way. Explicitly, for arbitrary \( (x_1, x_2) \in 30 \times 30 \),

\[
p_y(x_1, x_2) = \begin{cases} 
1 & \text{if } x_1 = y_1 \text{ and } x_2 = y_2 \\
\frac{3}{4} & \text{if } (x_1 = y_1 \text{ and } (x_2 = y_2 + 1 \text{ or } x_2 = y_2 - 1)) \\
& \quad \text{or } (x_2 = y_2 \text{ and } (x_1 = y_1 + 1 \text{ or } x_1 = y_1 - 1)) \\
\frac{1}{2} & \text{if } (x_1 = y_1 - 1 \text{ and } (x_2 = y_2 + 1 \text{ or } x_2 = y_2 - 1)) \\
& \quad \text{or } (x_1 = y_1 + 1 \text{ and } (x_2 = y_2 + 1 \text{ or } x_2 = y_2 - 1)) \\
0 & \text{otherwise.}
\end{cases}
\]
Figure 4.3 Four examples of data in a plane on the left and their classification into three groups (yellow, blue, red) on the right. The third row contains some errors, given by yellow dots in a region that should be entirely red. The last row describes a failed classification.
The probabilities around the crossed point $y$ incorporate the idea of distance to $y$. In this way we can organise the data as a multiset of predicates, represented in the sequel as $\Psi \in \mathcal{N}(\text{Pred}(30 \times 30))$, with a predicate $p_y$ for each crossed point $y$.

We now come to a crucial decision, namely which form of EM to use: M-learning, or C-learning, or some form of MC-learning. Experimenting with this example shows that none of these approaches work well if we start from an arbitrary state $\omega \in \mathcal{D}(X)$ and channel $e: X \rightarrow Y$ and iterate towards improvements: this often leads to failed classifications like in the last row of Figure 4.3. Things improve if we ‘help’ the channel a little bit, by incorporating initially information about where the clusters are — while still choosing the initial state $\omega$ randomly. The learning approach that performs best, experimentally, is the form of MC-learning that we describe next.

**Proposition 4.10.1.** Let $\sigma \in \mathcal{D}(X)$ and $c: X \rightarrow Y$ be given, for finite sets $X, Y$, together with data $\Phi \in \mathcal{N}(\text{Fact}(Y))$. Consider these data as a multiset of point-data $\bar{\Phi} \in \mathcal{N}(\mathcal{M}(Y))$, via Remark 2.1.6 where:

$$\bar{\Phi} := \sum_{\rho} \Phi(\rho) \mid \sum_{\rho} p(\rho)(y) \ .$$

Using MC-learning along channel $c$ with $\bar{\Phi}$ yields, according to Theorem 4.9.7 as better state:

$$\sigma' = \sum_{\rho} \frac{\Phi(\rho)}{|\mathcal{D}|} \cdot \text{Clrn}(\sigma, e, \sum_{\rho} p(y|y)) = \sum_{\rho} \frac{\Phi(\rho)}{|\mathcal{D}|} \cdot \sigma|_{\{\rho|_{\langle \tau \mid \{1 \rangle} = \tau \}}^{|c, \{\tau \mid \{1 \rangle} = \tau \}}$$

with better channel $c' = \tau[0,1] \mid 1,0] \in \mathcal{M}(X \times Y)$ is the multiset:

$$\tau := \sum_{\rho} \frac{\Phi(\rho)}{|\mathcal{D}|} \cdot \sigma|_{\{\rho|_{\langle \tau \mid \{1 \rangle} = \tau \}}^{|c, \{\tau \mid \{1 \rangle} = \tau \}} \otimes \left( \sum_{\rho} p(y|y) \right).$$

**Proof.** The equation for $\sigma'$ follows directly from the formula for $\omega'$ in (4.30), where we use that for point-data $\varphi \in \mathcal{M}(Y)$ one has $\text{Clrn}(\sigma, e, \varphi) = \sigma|_{\{\rho|_{\langle \tau \mid \{1 \rangle} = \tau \}}^{|c, \{\tau \mid \{1 \rangle} = \tau \}}$, see (4.18).

The above description for $c'$ matches $e'$ in (4.30) since:

$$c'(x)(y) = \frac{\tau(x, y)}{\sum_{z} \tau(x, z)} = \frac{\sum_{\rho} \Phi(\rho) \cdot \text{Clrn}(\sigma, e, \sum_{\rho} p(y|y))(x) \cdot p(y)}{\sum_{\rho} \Phi(\rho) \cdot \sum_{\rho} p(y|y)(x) \cdot p(y)}.$$

We finish by describing how we get the classification results in the column on the right in Figure 4.3. The starting point is the data $\Psi \in \mathcal{M}(\text{Pred}(30 \times 30))$ as described in the beginning of this section. We then pick an arbitrary $\omega \in \mathcal{D}(Y, R, B)$ as initial state. A channel $e: \{Y, R, B\} \rightarrow 30 \times 30$ is chosen by
picking three distributions $e(Y)$, $e(R)$, $e(B)$ on the $30 \times 30$ grid. We then iterate 25 computations of better $\omega'$ and $e'$ according to Proposition 4.10.1. Choosing the channel $e$ completely randomly leads to many failed classification results, because the iterations get stuck in a local optimum. We need to point them in the right direction by ‘helping’ the channel $e$ a little bit. We can do so by mixing the three cluster centers — chosen at the very beginning of each experiment — into the choices of the three states $e(Y)$, $e(R)$, $e(B)$. One can steer the amount of randomness in this mix via a scaling factor. The more randomness, the poorer the classification results.

What we conclude at the end of this chapter is that in a general learning situation with a channel $X \rightarrow Y$, with data on $Y$, with the methods that we have discussed:

- learning only the ‘latent’ state on $X$, when the channel $X \rightarrow Y$ is given (and fixed), works well;
- learning both the state on $X$ and the channel $X \rightarrow Y$ works poorly in general, since the iterations can easily go off in a wrong direction. Still, one can get acceptable results if the channel is initially not chosen randomly but already fits the situation at hand in a reasonable manner.
References

Chapter 4. References


