Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 347 (2019) 143–160

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Learning along a Channel: the Expectation part of Expectation-Maximisation

Bart Jacobs[1]

Institute for Computing and Information Sciences (iCIS)
Radboud University Nijmegen
The Netherlands

Abstract

This paper first investigates a form of frequentist learning that is often called Maximal Likelihood Estimation (MLE). It is redescribed as a natural transformation from multisets to distributions that commutes with marginalisation and disintegration. It forms the basis for the next, main topic: learning of hidden states, which is reformulated as learning along a channel. This topic requires a fundamental look at what data is and what its validity is in a particular state. The paper distinguishes two forms, denoted as 'M' for 'multiple states' and 'C' for 'copied states'. It is shown that M and C forms exist for validity of data, for learning from data, and for learning along a channel. This M/C distinction allows us to capture two completely different examples from the literature which both claim to be instances of Expectation-Maximisation.

Keywords: Probabilistic learning, Maximal Likelihood Estimation, latent variables, Expectation-Maximisation, learning along a channel

1 Introduction

Bayesian networks are graphical models for efficiently organising probabilistic information [1,3,8,15,16,17]. These models can be used for probabilistic reasoning (inference), where the updated probability is inferred from certain evidence. These techniques are extremely useful, for instance in a medical setting, where symptoms and measurements can be used as evidence, and the inferred probability can help a doctor reach a decision.

A basic question is how to obtain accurate Bayesian networks. This question involves two parts: how to determine the underlying graph structure, and how to obtain the probabilities in the conditional probability tables (CPTs) of the network. The first part is called *structure learning*, and the second part is called *parameter*

[1] Email: bart@cs.ru.nl

https://doi.org/10.1016/j.entcs.2019.09.008

learning. Here we concentrate on the latter, especially for discrete probability distributions.

One way of obtaining the parameters of Bayesian network is to learn them from experts. However, it is more efficient and cheaper to learn the parameters from data, if available. The data is typically organised in (very large) tables; we shall describe such tables, say with n dimensions, as n-ary multisets in $\mathcal{M}(X_1 \times \cdots \times X_n)$, where $\mathcal{M}$ is the multiset monad on the category of sets. Frequentist learning from a multiset happens by counting and normalising. It is described here as as a natural transformation of the form $\mathcal{M}_* \Rightarrow \mathcal{D}$, where $\mathcal{M}_*$ is the (sub)monad of *non-empty* multisets, and where $\mathcal{D}$ is the discrete probability distribution monad. This learning technique is called *maximal likelihood estimation* (MLE), see *e.g.* [8, Ch.17], [16, §17.1] or [15, §6.1.1].

This paper takes a fresh, fundamental look at learning. It starts from the basic notion of validity, or expected value, $\omega \models p$ of a predicate p in a state (distribution) ω. This validity is a number in the unit interval $[0, 1]$. In its basic form, learning is described as increasing this validity by changing the state ω, while keeping the predicate/evidence p fixed. Thus learning involves changing ω into ω' such that the validity $\omega' \models p$ is greater than $\omega \models p$. One ways of doing this is taking $\omega' = \omega|_p$, where $\omega|_p$ is the state ω updated with (conditioned by) p, see below for details.

In this paper we go beyond the validity of a single predicate. We abstractly describe data, as source of learning, in terms of a multiset ψ of predicates. It turns out that there are two different forms of validity for such data, which we will describe as *multiple-state* validity $\omega \models\!\!\!\models_{\mathrm{M}} \psi$ and *copied-state* validity $\omega \models\!\!\!\models_{\mathrm{C}} \psi$. The distinction corresponds to the well-known situation in probability theory where one has an urn with coloured balls and one draws with replacement (multiple-state) or without replacement (copied-state). Accordingly, there are two forms of learning, namely *M-learning* which increases M-validity $\models\!\!\!\models_{\mathrm{M}}$, and *C-learning* which increase C-validity $\models\!\!\!\models_{\mathrm{C}}$.

In a next step we look at learning of 'latent' or 'hidden' variables on a probability space X, which is only indirectly accessible. In the current setting this means that we have data, not on X itself, but on a different space Y, with a channel (Kleisli map) $X \to\!\!\!\!\circ\; Y$ between them. Obtaining a newly learned state ω' on X, from a given state ω, in this situation is the 'Expectation' part of the Expectation-Maximisation (EM) learning method [9]. We will identify two approaches, again denoted as M and C, to such learning along a channel.

A key discovery of this paper is that this M/C distinction explains two completely different approaches to learning along a channel in the literature, in classic references [18] and in [10], which are nevertheless both called Expectation-Maximisation.

The paper starts with mathematical preliminaries about states, channels, predicates and conditioning in Section 2 and with a concrete illustration of learning from tabular data, which is abstracted to a 'frequentist learning' natural transformation *Flrn* from multisets to distributions in Section 3. The paper continues in Section 4 with the identification of data in terms of multisets of predicates, together

with associated forms of M/C-validity and M/C-learning. This is approach is extended to learning along a channel in Section 5, where the main examples from the literature [10,18] are redescribed in the new M/C-framework.

Proofs of the main learning results involve some elementary real analysis and are relegated to the appendix. The calculations in this paper have been carried out with the EfProb library [5] for channel-based probability.

2 Mathematical preliminaries

A multiset is a 'set' in which (finitely many) elements may occur multiple times, with non-negative real numbers, in $\mathbb{R}_{\geq 0}$, as multiplicities. We write $\mathcal{M}(X)$ for the set of such multisets over a set X, defined as:

$$\mathcal{M}(X) \coloneqq \{\phi \colon X \to \mathbb{R}_{\geq 0} \mid \mathit{supp}(\phi) \text{ is finite}\},$$

where $\mathit{supp}(\phi)$ is the support of ϕ, *i.e.* the subset $\{x \in X \mid \phi(x) \neq 0\}$. We shall write $\mathcal{N}(X) \subseteq \mathcal{M}(X)$ for the subset of *natural* multisets, where $\phi \in \mathcal{N}(X)$ if $\phi(x) \in \mathbb{N}$ for each x. We often write concrete multisets as finite formal sums, using a 'ket' notation: $\phi = \sum_x \phi(x)|x\rangle$. Taking multisets on a set is functorial: for $f \colon X \to Y$ we get $\mathcal{M}(f) \colon \mathcal{M}(X) \to \mathcal{M}(Y)$ via $\mathcal{M}(f)(\phi)(y) = \sum_{x \in f^{-1}(y)} \phi(x)$. Alternatively, for formal sums: $\mathcal{M}(f)(\sum_i r_i|x_i\rangle) = \sum_i r_i|f(x_i)\rangle$. In fact, $\mathcal{M}$ is a monad on the category of sets, but this is not really needed here.

A *probability distribution* or a *multinomial* or a *state* is a multiset whose multiplicities add up to one. We define the subset of those as:

$$\begin{aligned}\mathcal{D}(X) &\coloneqq \{\phi \in \mathcal{M}(X) \mid \textstyle\sum_x \phi(x) = 1\} \\ &= \{\phi \colon X \to [0,1] \mid \mathit{supp}(\phi) \text{ is finite, and } \textstyle\sum_x \phi(x) = 1\}.\end{aligned}$$

This $\mathcal{D}$ is also monad on sets.

A *channel* $f \colon X \mathrel{-\!\!\circ\!\!\to} Y$ is a probabilistic computation from X to Y. It is a 'Kleisli' map $f \colon X \to \mathcal{D}(Y)$. Such a channel can 'push' a state $\omega \in \mathcal{D}(X)$ forward to a state $f \gg \omega \in \mathcal{D}(Y)$, via 'Kleisli extension' or 'state transformation', where $(f \gg \omega)(y) = \sum_x \omega(x) \cdot f(x)(y)$. Via $\gg$ we can define composition $g \circ f$ of channels as $(g \circ f)(x) = g \gg f(x)$.

A predicate on a set X is a function $p \colon X \to [0,1]$. Specifically, each subset (event) $E \subseteq X$ gives rise to a 'sharp' predicate $\mathbf{1}_E \colon X \to [0,1]$ by $\mathbf{1}_E(x) = 1$ when $x \in E$ and $\mathbf{1}_E(x) = 0$ when $x \notin E$. For an element $x \in X$ we whall write $\mathbf{1}_x$ for $\mathbf{1}_{\{x\}}$ and call $\mathbf{1}_x$ a point-predicate. We shall write $\mathrm{Pred}(X) = [0,1]^X$ for the set of all predicates on X.

The validity $\omega \models p$ in $[0,1]$ of a predicate $p \in [0,1]^X$ in state $\omega \in \mathcal{D}(X)$ is defined as the expected value defined on the left below.

$$\omega \models p \;\coloneqq\; \textstyle\sum_x \omega(x) \cdot p(x) \qquad\qquad \omega|_p(x) \;\coloneqq\; \dfrac{\omega(x) \cdot p(x)}{\omega \models p}.$$

When this validity $\omega \models p$ is non-zero, one can update (condition) ω with predicate p, to get a new state $\omega|_p \in \mathcal{D}(X)$, as defined above, on the right.

Given a channel $c\colon X \nrightarrow Y$, one can transform a predicate $q\colon Y \to [0,1]$ on its codomain into predicate $c \ll q\colon X \to [0,1]$ on its domain, via: $(c \ll q)(x) = \sum_y c(x)(y) \cdot q(y)$. The validities $\omega \models c \ll q$ and $c \gg \omega \models q$ are then equal.

We are now in a position to describe a basic form of learning, in a situation with a validity expression:

$$\begin{array}{ccc} & \omega \models p & \\ \text{distribution / state} & & \text{predicate / evidence} \end{array} \tag{1}$$

Learning involves increasing this validity, by changing the state ω into a new state ω' such that $\omega' \models p \;\geq\; \omega \models p$. Thus, we don't change the predicate p; we consider it as given evidence that we need to adjust to. We do so by changing the state ω so that it better fits the evidence.

The next result captures a basic intuition, namely that learning can happen via conditioning. The proof is postponed to the appendix.

Theorem 2.1 *Updating with p increases the validity of p, as in: $\omega \models p \;\leq\; \omega|_p \models p$.*

In this paper we concentrate on such single steps in learning and not on their iteration. Convergence, either in an order-theoretic or in a metric sense (see [13]), is out of scope.

We still need the basic notion of product, both for states and for predicates. For two states $\sigma \in \mathcal{D}(X)$ and $\tau \in \mathcal{D}(Y)$ on sets X, Y one can form the product state $\sigma \otimes \tau \in \mathcal{D}(X \times Y)$ via $(\sigma \otimes \tau)(x,y) = \sigma(x) \cdot \tau(y)$.

For predicates there are two different products/conjunctions, namely parallel conjunction $p \otimes q$ and sequential conjunction $p_1 \,\&\, p_2$.

(i) For two predicates $p \in [0,1]^X$ and $q \in [0,1]^Y$ on *different* sets X, Y there is a parallel product $p \otimes q \in [0,1]^{X \times Y}$ defined by $(p \otimes q)(x,y) = p(x) \cdot q(y)$. It is easy to see that then: $\sigma \otimes \tau \models p \otimes q$ equals $(\sigma \models p) \cdot (\tau \models q)$.

(ii) For two predicates $p_1, p_2 \in [0,1]^X$ on the *same* set X we can form the sequential product $p_1 \,\&\, p_2 \in [0,1]^X$ on this same X via: $(p_1 \,\&\, p_2)(x) = p_1(x) \cdot p_2(x)$. Then, using the diagonal map/channel Δ,

$$\omega \models p_1 \,\&\, p_2 \;=\; \omega \models \Delta \ll (p_1 \otimes p_2) \;=\; \Delta \gg \omega \models p_1 \otimes p_2 \text{ where } \Delta \gg \omega \neq \omega \otimes \omega. \tag{2}$$

Sequential conjunction & allows us to reduce successive state updates to a single update, since one has:

$$\big(\omega|_p\big)|_q = \omega|_{p \& q} = \omega|_{q \& p} = \big(\omega|_q\big)|_p. \tag{3}$$

For more information, see [11,14].

3 Tables and distributions

This section will elaborate a simple example in order to provide background information about the setting for learning. Consider the table (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:

	no medicine	**medicine 1**	**medicine 2**	**totals**
high	10	35	25	70
low	5	10	15	30
totals	15	45	40	100

(4)

We consider several ways to 'learn' from this table.

(1) We can form the cartesian product $\{H, T\} \times \{0, 1, 2\}$ of the possible outcomes and then capture the above table as a multiset over this product:

$$\tau \coloneqq 10|H, 0\rangle + 35|H, 1\rangle + 25|H, 2\rangle + 5|L, 0\rangle + 10|L, 1\rangle + 15|L, 2\rangle. \tag{5}$$

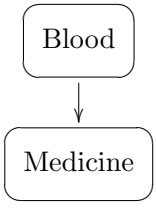
We can normalise this multiset τ. It yields a joint probability distribution:

$$\omega \coloneqq 0.10|H, 0\rangle + 0.35|H, 1\rangle + 0.25|H, 2\rangle + 0.05|L, 0\rangle + 0.10|L, 1\rangle + 0.15|L, 2\rangle \tag{6}$$

Such a distribution, directly derived from a table, is sometimes called an *empirical* distribution [8].

(2) The first and second marginals $\mathsf{M}_1(\omega) \coloneqq \mathcal{D}(\pi_1)(\omega)$ and $\mathsf{M}_2(\omega) \coloneqq \mathcal{D}(\pi_2)(\omega)$ of this joint probability distribution ω capture the blood pressure probabilities and the medicine probabilities separately, as: $\mathsf{M}_1(\omega) = 0.7|H\rangle + 0.3|L\rangle$ and $\mathsf{M}_2(\omega) = 0.15|0\rangle + 0.45|1\rangle + 0.4|2\rangle$. These marginal distributions can also be obtained directly from the above table (4), via the normalisation of 'totals' column and row. This fact looks like a triviality, but involves a naturality property (see Lemma 3.2 below).

(3) Next we wish to use the above table (4) to learn the parameters (table entries) for the simple Bayesian network on the right. We then need to fill in the associated conditional probability tables. These entries are obtained from the last column in Table 4, for the initial blood distribution $0.7|H\rangle + 0.3|L\rangle$, and from the two rows in the table; the latter yield two distributions for medicine usage, via normalisation.



H	L
0.7	0.3

	0	1	2
H	$1/7$	$1/2$	$5/14$
L	$1/6$	$1/3$	$1/2$

(4) In the categorical look at Bayesian networks (see *e.g.* [12,14]) these conditional probability tables correspond to *channels*: Kleisli maps for the distribution monad $\mathcal{D}$. In the above case, the channel $c\colon \{H, T\} \nrightarrow \{0, 1, 2\}$ corresponding to the

medicine table in the previous point is:

$$c(H) = \tfrac{1}{7}|0\rangle + \tfrac{1}{2}|1\rangle + \tfrac{5}{14}|2\rangle \qquad c(L) = \tfrac{1}{6}|0\rangle + \tfrac{1}{3}|1\rangle + \tfrac{1}{2}|2\rangle.$$

The second marginal $\mathsf{M}_2(\omega)$ then equals $c \gg \mathsf{M}_1(\omega)$.

(5) Given a joint distribution $P(x, y)$ there is a standard way to extract a channel $P(y \mid x)$ by taking conditional probabilities. This process is often called *disintegration*, and is studied systematically in [6,7]. If we disintegrate the above distribution ω on the product $\{H, T\} \times \{0, 1, 2\}$ we obtain as channel $\{H, T\} \nrightarrow \{0, 1, 2\}$, precisely the map c from the previous point — obtained in point **(3)** directly via the Table (4). This is a highly relevant property, which essentially means that (this kind of) learning can be done locally.

3.1 Frequentist learning by counting

The above example illustrates how probabilistic information can be extracted from a table with numeric data — in a frequentist manner — essentially by counting. This will now be described in terms of a natural transformation *Flrn* from multisets to distributions, as used in the passage from the multiset τ in (5) to the distribution $\omega = \mathit{Flrn}(\tau)$ in (6). As mentioned in the introduction, maximal likelihood estimation (MLE) is one kind of parameter learning, see *e.g.* [8,15,16]. Our categorical reformulation for discrete probability distributions (multinomials) uses the non-empty multiset functor $\mathcal{M}_*$ and the distribution functor $\mathcal{D}$ from Section 2. It turns out that the process of learning-by-counting involves some basic categorical structure: it is a monoidal natural transformation, that can be applied locally.

Definition 3.1 *For a set X, let $\mathcal{M}_*(X) \subseteq \mathcal{M}(X)$ be the subset of non-empty (i.e. non-null) multisets. We define (discrete) maximal likelihood estimation as the normalisation function* Flrn$\colon \mathcal{M}_*(X) \to \mathcal{D}(X)$*, determined by:*

$$\mathit{Flrn}(\phi)(x) \coloneqq \frac{\phi(x)}{|\phi|} \quad \text{where} \quad |\phi| \coloneqq \textstyle\sum_y \phi(y), \text{ i.e. } \mathit{Flrn}\big(\sum_i r_i|x_i\rangle\big) = \sum_i \frac{r_i}{\sum_j r_j}|x_i\rangle. \tag{7}$$

Lemma 3.2 *The maps* Flrn$\colon \mathcal{M}_*(X) \to \mathcal{D}(X)$ *form a natural transformation $\mathcal{M}_* \Rightarrow \mathcal{D}$. This natural transformation is monoidal, but not a map of monads.*

Proof. We only prove naturality. For a function $h\colon X \to Y$,

$$\begin{aligned}\big(\mathit{Flrn} \circ \mathcal{M}(h)\big)\big(\textstyle\sum_i r_i|x_i\rangle\big) &= \mathit{Flrn}\big(\textstyle\sum_i r_i|h(x_i)\rangle\big) \\ &= \textstyle\sum_i \frac{r_i}{r}|h(x_i)\rangle \qquad \text{for } r = \sum_i r_i \\ &= \mathcal{D}(h)\big(\textstyle\sum_i \frac{r_i}{r}|x_i\rangle\big) \;=\; \big(\mathcal{D}(h) \circ \mathit{Flrn}\big)\big(\textstyle\sum_i r_i|x_i\rangle\big). \qquad \square\end{aligned}$$

In Section 3 we mentioned that one can extract a conditional probability table (or channel) either directly from the table of data, or from the associated empirical probability distribution. In the first case one takes the obvious map $\mathcal{M}(X \times Y) \to$

$\mathcal{M}(X)^Y$. There is a similar 'disintegration' mapping $\mathcal{D}(X \times Y) \to \mathcal{D}(Y)^X$, turning a joint distribution into a channel; it involves an additional normalisation step and (thus) a side-condition, see [6]. It is not hard to show that the learning maps $Flrn$ commute with these two extraction maps, for multisets and for distributions. This is a fundamental result, since it says that distributions can be obtained locally from a table, as illustrated in Section 3. Nevertheless, we will not elaborate it in detail. Instead, we show later in Proposition 4.3 that $Flrn(\phi)$ is optimal, in a suitable sense.

4 Data, validity, and learning

So far we have described learning with a single predicate. In practice learning happens from 'data', so the first question we should address is: what is data? It often comes in the form of sequences or (multidimentional) tables. The order of data items does not matter for updating, see (3), but multiple occurrences of data items are relevant. These two aspects suggest that data items on a set X should be organised in the form of multisets in $\mathcal{N}(X)$. However, a more powerful perspective uses multisets of predicates as data. A data item, in the form of a predicate, may be a point-predicate, when there is certainty about the element involved. But when some data is missing, or when there is uncertainty, we may use a uniform predicate. Hence we shall use elements $\Phi \in \mathcal{M}(\mathrm{Pred}(X))$ as data on X, with what we call *point-data* $\varphi \in \mathcal{M}(X)$ as special case, corresponding to data with point-predicates $\mathbf{1}_x$ for $x \in X$.

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.

(i) What is the probability of getting two heads? Most people will say $\frac{1}{4}$. The idea behind this answer is that we flip the coint 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 σ is used twice, as in:

$$\sigma \otimes \sigma \models \mathbf{1}_H \otimes \mathbf{1}_H = \big(\sigma \models \mathbf{1}_H\big) \cdot \big(\sigma \models \mathbf{1}_H\big) = \tfrac{1}{2} \cdot \tfrac{1}{2} = \tfrac{1}{4}.$$

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, truthfull) 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:

$$\sigma \models \mathbf{1}_H \,\&\, \mathbf{1}_H = \sigma \models \mathbf{1}_H = \tfrac{1}{2}.$$

(ii) 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:

$$\sigma \otimes \sigma \models \mathbf{1}_H \otimes \mathbf{1}_T = \big(\sigma \models \mathbf{1}_H\big) \cdot \big(\sigma \models \mathbf{1}_T\big) = \tfrac{1}{2} \cdot \tfrac{1}{2} = \tfrac{1}{4}.$$

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:

$$\sigma \models \mathbf{1}_H \mathbin{\&} \mathbf{1}_T = \sigma \models \mathbf{0} = 0.$$

(iii) Next assume it is dark and the observers cannot see the coin very clearly. We consider the predicate $p = 0.8 \cdot \mathbf{1}_H + 0.2 \cdot \mathbf{1}_T$ giving 80% certainty for head, and $q = 0.7 \cdot \mathbf{1}_H + 0.3 \cdot \mathbf{1}_T$ with only 70% certainty. What is now the probability of seeing p and q? The multiple-state perspective gives:

$$\sigma \otimes \sigma \models p \otimes q = \big(\sigma \models p\big) \cdot \big(\sigma \models q\big) = \big(\tfrac{1}{2} \cdot 0.8 + \tfrac{1}{2} \cdot 0.2\big) \cdot \big(\tfrac{1}{2} \cdot 0.7 + \tfrac{1}{2} \cdot 0.3\big) = \tfrac{1}{4}.$$

The alternative perspective now gives:

$$\sigma \models p \mathbin{\&} q = \sigma \models 0.56 \cdot \mathbf{1}_H + 0.06 \cdot \mathbf{1}_T = \tfrac{1}{2} \cdot 0.56 + \tfrac{1}{2} \cdot 0.06 = 0.31.$$

We have referred to the first approach with tensor $\sigma \otimes \sigma$ of states as the *multiple-state* perspective. We will refer to the second approach as the *copied-state* perspective, in the light of (2). These different perspectives lead to different notions of validity of data.

Definition 4.1 Let $\omega \in \mathcal{D}(X)$ be a state and $\Phi \in \mathcal{M}\big(\mathrm{Pred}(X)\big)$ be data on X.

(i) The *multiple-state validity*, or M-validity for short, of Φ in ω is defined as:

$$\omega \models\!\!\!\models_{\mathrm{M}} \Phi \coloneqq \prod\nolimits_p \big(\omega \models p\big)^{\Phi(p)}. \tag{8}$$

(ii) The *copied-state validity*, or C-validity, of Φ in ω is:

$$\omega \models\!\!\!\models_{\mathrm{C}} \Phi \coloneqq \omega \models \mathop{\&}\nolimits_p p^{\Phi(p)}. \tag{9}$$

When Φ consists of point-data, these validities become $\prod_{x \in supp(\Phi)} \omega(x)^{\Phi(x)}$ and $\omega \models \mathop{\&}_{x \in supp(\Phi)} \mathbf{1}_x$, respectively.

For clarity: p^n in point (ii) is the n-fold conjunction $p \mathbin{\&} \cdots \mathbin{\&} p$, with $p^0 = \mathbf{1}$. When p is sharp, then $p^n = p$, for $n > 1$. In particular, point-predicates $\mathbf{1}_x$ are sharp.

Corresponding to M/C-validity, there is M/C-learning, where the aim is, as before, to modify the state in order to increase validity. The next result gives the essentials. The proof (of the first point) is in the appendix.

Theorem 4.2 *Let $\omega \in \mathcal{D}(X)$ be a state with data $\Phi \in \mathcal{M}(\mathrm{Pred}(X))$.*

(i) *M-learning $Mlrn(\omega, \Phi) \models\!\!\!\models_M \Phi \;\geq\; \omega \models\!\!\!\models_M \Phi$ can be done via the newly learned state:*

$$Mlrn(\omega, \Phi) \coloneqq \sum\nolimits_p \frac{\Phi(p)}{|\Phi|} \cdot \omega|_p \qquad \textit{where} \qquad |\Phi| \coloneqq \textstyle\sum_p \Phi(p).$$

(ii) *C-learning* $Clrn(\omega, \Phi) \models_C \Phi \;\geq\; \omega \models_C \Phi$ *can be done directly via Theorem 2.1, with* $Clrn(\omega, \Phi) = \omega\big|_{\&_p p^{\Phi(p)}}$.

When we apply point (i) to point-data $\varphi \in \mathcal{N}(X)$ we get as newly learned state:

$$Mlrn(\omega, \varphi) = \sum_x \frac{\varphi(x)}{|\varphi|} \cdot \omega|_{\mathbf{1}_x} = \sum_x \frac{\varphi(x)}{|\varphi|} |x\rangle = Flrn(\varphi).$$

Hence we rediscover frequentist learning *Flrn* from Section 3.1. 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.* [16, Ex. 17.5], which we reproduce here (with a proof in the appendix).

Proposition 4.3 *For non-empty point-data* $\varphi \in \mathcal{M}(X)$ *the predicate "M-validity of* φ*"*

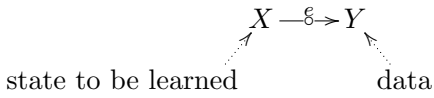
$$\mathcal{D}(X) \xrightarrow{(-) \models_M \varphi} [0, 1]$$

takes its maximum at the distribution $Flrn(\varphi) \in \mathcal{D}(X)$ *that is obtained by frequentist learning. This says:*

$$Flrn(\varphi) = \operatorname*{argmax}_\omega \omega \models_M \varphi.$$

5 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:



(10)

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. This forms the E-part of what is called Expectation-Maximisation (EM), see [9]. In the M-part the channel e becomes a learning goal in itself. In Expectation-Maximisation these E- and M-parts are alternated. But here we concentrate on the E-part only and assume that the channel e is given and remains fixed.

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 $\Psi \in \mathcal{N}\big(\mathrm{Pred}(Y)\big)$ on Y to data on X, via the channel $e \colon X \nrightarrow Y$. This can be done easily, by using that the multiset operation $\mathcal{N}$ is functorial and that predicate transformation yields a function $e \ll$

$(-)\colon \mathrm{Pred}(Y) \to \mathrm{Pred}(X)$. Thus we get data on X via a new data transformation, which we shall write as:

$$e \ll_{\mathcal{N}} \Psi \coloneqq \mathcal{N}\big(e \ll (-)\big)(\Psi) = \sum\nolimits_p \Psi(p)\big|\, e \ll p\,\big\rangle. \tag{11}$$

Definition 5.1 Let $e\colon X \nrightarrow Y$ be a channel with data $\Psi \in \mathcal{N}\big(\mathrm{Pred}(Y)\big)$ on its codomain. We define M/C-learning along e from Ψ as M/C-learning from $e \ll_{\mathcal{N}} \Psi$ in (11). Concretely this turns a state $\omega \in \mathcal{D}(X)$ into a newly learned states on X, where:

(i) for M-learning,

$$\mathit{Mlrn}(\omega, e, \Psi) \coloneqq \mathit{Mlrn}(\omega, e \ll_{\mathcal{N}} \Psi) = \sum\nolimits_p \frac{\Psi(p)}{|\Psi|} \cdot \omega|_{e \ll p}. \tag{12}$$

(ii) for C-learning,

$$\begin{array}{rcl} \mathit{Clrn}(\omega, e, \Psi) & \coloneqq & \mathit{Clrn}(\omega, e \ll_{\mathcal{N}} \Psi) = \omega\big|_{\&_p (e \ll p)^{\Psi(p)}} \\ & \overset{(3)}{=} & \omega|_{(e \ll p_1)^{n_1}} \cdots |_{(e \ll p_k)^{n_k}} \quad \text{when } \Psi = \sum_{1 \leq i \leq k} n_i |p_i\rangle. \end{array} \tag{13}$$

Notice that we have overloaded the notation *Mlrn* and *Clrn*, where its meaning depends on the number of arguments: two for ordinary learning, as in Theorem 4.2, and three for learning along a channel.

Before looking at examples, we show that there is an alternative description of M-learning along a channel for point-data in terms of the so-called 'dagger' or 'Bayesian inversion' of the channel, see [6,7]. Given a state $\omega \in \mathcal{D}(X)$ on the domain of a channel $e\colon X \nrightarrow Y$ we can turn e around to a channel $e^{\dagger}_{\omega}\colon Y \nrightarrow X$, via conditioning with a transformed point predicate:

$$e^{\dagger}_{\omega}(y) \coloneqq \omega|_{e \ll \mathbf{1}_y} = \sum_x \frac{\omega(x) \cdot e(x)(y)}{(e \gg \omega)(y)} \big|\, x \,\big\rangle. \tag{14}$$

This dagger satisfies all sorts of nice properties, see [6,7] and the references given there for more information. Here we concentrate on the relevance of the dagger of a channel in learning — for point-data.

Lemma 5.2 *For a state $\omega \in \mathcal{D}(X)$ and channel $e\colon X \nrightarrow Y$ with* point-data *$\varphi \in \mathcal{N}(X)$ we get:*

$$\mathit{Mlrn}(\omega, e, \varphi) = e^{\dagger}_{\omega} \gg \mathit{Flrn}(\varphi). \tag{15}$$

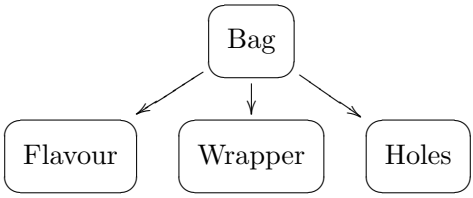
Proof. Since:

$$\begin{array}{rcl} \mathit{Mlrn}(\omega, e, \varphi) \overset{(12)}{=} \sum_x \frac{\varphi(x)}{|\varphi|} \cdot \omega|_{e \ll \mathbf{1}_x} & \overset{(14)}{=} & \sum_x \frac{\varphi(x)}{|\varphi|} \cdot e^{\dagger}_{\omega}(x) \\ & \overset{(7)}{=} & \sum_x \mathit{Flrn}(\varphi)(x) \cdot e^{\dagger}_{\omega}(x) \;=\; e^{\dagger}_{\omega} \gg \mathit{Flrn}(\varphi). \end{array}$$

□

We now come to the two classic examples in the literature that we will redescribe using the newly developed M/C distinction.

Example 5.3 The first example is taken from [18, §20.3], where it is used as an illustration of the Expectation-Maximisation (EM) algorithm. It involves the Bayesian network as below, with (two) bags of candies, described by three features, namely their flavour, their wrapper, and whether or not they have holes.



We number the bags as 0 and 1, so that we use the set $\{0,1\}$ for bags. The flavours can be cherry and lime, in the set $\{C, L\}$; the wrappers can be red or green, in $\{R, G\}$, and there can be a hole or not, in $\{H, H^\perp\}$. The three arrows in the Bayesian network correspond to three channels $f\colon \{0,1\} \nrightarrow \{C,L\}$, $w\colon \{0,1\} \nrightarrow \{R,G\}$, $h\colon \{0,1\} \nrightarrow \{H, H^\perp\}$, which have equal probabilities in [18]:

$$\begin{array}{lll} f(0) = \frac{6}{10}|C\rangle + \frac{4}{10}|L\rangle & w(0) = \frac{6}{10}|R\rangle + \frac{4}{10}|G\rangle & h(0) = \frac{6}{10}|H\rangle + \frac{4}{10}|H^\perp\rangle \\ f(1) = \frac{4}{10}|C\rangle + \frac{6}{10}|L\rangle & w(1) = \frac{4}{10}|R\rangle + \frac{6}{10}|G\rangle & h(1) = \frac{4}{10}|H\rangle + \frac{6}{10}|H^\perp\rangle. \end{array}$$

These three channels are combined into a single (three-)tuple channel $\langle f, w, h\rangle\colon \{0,1\} \nrightarrow \{C,L\} \times \{R,G\} \times \{H, H^\perp\}$. At 0 it is:

$$\begin{array}{rcl} \langle f, w, h\rangle(0) & = & f(0) \otimes w(0) \otimes h(0) \\ & = & \frac{216}{1000}|C,R,H\rangle + \frac{144}{1000}|C,R,H^\perp\rangle + \frac{144}{1000}|C,G,H\rangle + \frac{96}{1000}|C,G,H^\perp\rangle \\ & & + \frac{144}{1000}|L,R,H\rangle + \frac{96}{1000}|L,R,H^\perp\rangle + \frac{96}{1000}|L,R,H\rangle + \frac{64}{1000}|L,R,H^\perp\rangle. \end{array}$$

The point-data $\psi \in \mathcal{M}\big(\{C,L\} \times \{R,G\} \times \{H, H^\perp\}\big)$ is given by the multiset:

$$\begin{array}{rcl} \psi & = & 273|C,R,H\rangle + 93|C,R,H^\perp\rangle + 104|C,G,H\rangle + 90|C,G,H^\perp\rangle \\ & & + 79|L,R,H\rangle + 100|L,R,H^\perp\rangle + 94|L,R,H\rangle + 167|L,R,H^\perp\rangle. \end{array}$$

We thus have a pattern as described in (10), concretely of the form:

$$\{0,1\} \xrightarrow{\langle f,w,h\rangle} \{C,L\} \times \{R,G\} \times \{H, H^\perp\}$$

state to be learned — data ψ

We are now set for M-learning along the tuple channel $\langle f, w, h\rangle$ from these point-data on its codomain, via the dagger of the tuple channel, see Lemma 5.2. In [18]

this uses a prior distribution $\rho = \frac{6}{10}|0\rangle + \frac{4}{10}|1\rangle$. We thus compute the newly learned distribution on $\{0,1\}$ as:

$$Mlrn(\rho, \langle f, w, h\rangle, \psi) \overset{(15)}{=} \langle f, w, h\rangle^{\dagger}_{\rho} \gg Flrn(\psi) = 0.6124|0\rangle + 0.3876|1\rangle.$$

This probability 0.6124 is exactly as computed in [18, §20.3], but without the (dagger) channel machinery. Moreover, the 'multiple-state' versus 'copied-state' perspective does not occur there.

Our second example from [10] also claims to be an instance of expectation-maximisation. However, what happens there is completely different from expectation-maximisation as used in [18]. A crucial point of this paper is that the difference can be explained in terms of M-learning versus C-learning along a channel.

Example 5.4 The leading illustration of [10] 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 (16) below. There is a channel $e\colon \{0,1\} \nrightarrow \{H,T\}$ that captures two coins, with slightly different biases:

$$e(0) = \tfrac{3}{5}|H\rangle + \tfrac{2}{5}|T\rangle \qquad \text{and} \qquad e(1) = \tfrac{1}{2}|H\rangle + \tfrac{1}{2}|T\rangle.$$

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 $r|0\rangle + (1-r)|1\rangle$ 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 new learning action, independent from the outcomes of earlier lines.

point-data ψ	**C-learning**	**M-learning**
$5\|H\rangle + 5\|T\rangle$	$0.4491\|0\rangle + 0.5509\|1\rangle$	$0.4949\|0\rangle + 0.5051\|1\rangle$
$9\|H\rangle + 1\|T\rangle$	$0.805\|0\rangle + 0.195\|1\rangle$	$0.5354\|0\rangle + 0.4646\|1\rangle$
$8\|H\rangle + 2\|T\rangle$	$0.7335\|0\rangle + 0.2665\|1\rangle$	$0.5253\|0\rangle + 0.4747\|1\rangle$
$4\|H\rangle + 6\|T\rangle$	$0.3522\|0\rangle + 0.6478\|1\rangle$	$0.4848\|0\rangle + 0.5152\|1\rangle$
$7\|H\rangle + 3\|T\rangle$	$0.6472\|0\rangle + 0.3528\|1\rangle$	$0.5152\|0\rangle + 0.4848\|1\rangle$

(16)

This table is obtained via the formulas (13) and (12). M-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 [10], but uses higher precision. The term C-learning or the copied-state perspective do not

occur in [10]: the explanation of learning via Expectation-Maximisation given there consists basically of a single example.

We thus notice in these Examples 5.3 and 5.4 that [10] and [18] perform completely different computations in their explanation of expectation-maximisation, using, respectively, in the terminology of our setting, M-learning and C-learning along a channel. It is unclear why they do different things and still use the same Expectation-Maximisation terminology.

We conclude this section with some general observations. The most interesting one 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 5.5 (i) *Data transformation* $\ll_{\mathcal{N}}$ *from* (11) *is functorial in the sense that:*

$$\mathrm{id} \ll_{\mathcal{N}} \Psi = \Psi \qquad \text{and} \qquad (d \circ e) \ll_{\mathcal{N}} \Psi = e \ll_{\mathcal{N}} \big(d \ll_{\mathcal{N}} \Psi\big).$$

(ii) *M-learning is sequentially compositional in the sense that it interacts well with channel composition:*

$$\mathit{Mlrn}(\omega, \mathrm{id}, \Psi) = \mathit{Mlrn}(\omega, \Psi) \qquad \mathit{Mlrn}(\omega, d \circ e, \Psi) = \mathit{Mlrn}(\omega, d, e \ll_{\mathcal{N}} \Psi).$$

(iii) *C-learning is additively compositional, in the sense that:*

$$\mathit{Clrn}(\omega, e, \Phi + \Psi) = \mathit{Clrn}\big(\mathit{Clrn}(\omega, e, \Phi), e, \Psi\big).$$

Proof. (i) We use that predicate transformation is functorial, so that:

$$\mathrm{id} \ll (-) = \mathrm{id} \qquad \text{and} \qquad (d \circ e) \ll (-) = \big(e \ll (-)\big) \circ \big(d \ll (-)\big).$$

The result then follows by functoriality of taking multisets $\mathcal{N}$.

(ii) By the previous point:

$$\mathit{Mlrn}(\omega, \mathrm{id}, \Psi) = \mathit{Mlrn}(\omega, \mathrm{id} \ll_{\mathcal{N}} \Psi) = \mathit{Mlrn}(\omega, \Psi).$$

And:

$$\begin{aligned} \mathit{Mlrn}(\omega, d \circ e, \Psi) &= \mathit{Mlrn}(\omega, (d \circ e) \ll_{\mathcal{N}} \Psi) = \mathit{Mlrn}(\omega, e \ll_{\mathcal{N}} (d \ll_{\mathcal{N}} \Psi)) \\ &= \mathit{Mlrn}(\omega, e, d \ll_{\mathcal{N}} \Psi). \end{aligned}$$

(iii) Let's use the *ad hoc* notation:

$$e \lll \Psi \coloneqq \&_p \, (e \ll p)^{\Psi(p)}.$$

We then have $e \lll (\Phi + \Psi) = (e \lll \Phi) \mathbin{\&} (e \lll \Psi)$ since:

$$\begin{aligned} e \lll (\Phi + \Psi) \;= \mathop{\&}_p (e \ll p)^{(\Phi+\Psi)(p)} &= \mathop{\&}_p (e \ll p)^{\Phi(p)+\Psi(p)} \\ &= \mathop{\&}_p (e \ll p)^{\Phi(p)} \mathbin{\&} (e \ll p)^{\Psi(p)} \\ &= \mathop{\&}_p (e \ll p)^{\Phi(p)} \mathbin{\&} \mathop{\&}_p (e \ll p)^{\Psi(p)} \\ &= \big(e \lll \Phi\big)(x) \mathbin{\&} \big(e \lll \Psi\big)(x). \end{aligned}$$

Now we are almost done:

$$\begin{aligned} \mathit{Clrn}(\omega, e, \Phi + \Psi) \overset{(13)}{=} \omega|_{e \lll (\Phi+\Psi)} &= \omega|_{(e \lll \Phi) \& (e \lll \Psi)} && \text{as just shown} \\ &\overset{(3)}{=} \omega|_{e \lll \Phi}|_{e \lll \Psi} \\ &\overset{(13)}{=} \mathit{Clrn}(\omega, e, \Phi)|_{e \lll \Psi} \\ &\overset{(13)}{=} \mathit{Clrn}\big(\mathit{Clrn}(\omega, e, \phi), e, \Psi\big). \end{aligned}$$

□

6 Conclusions and further work

This paper has developed a systematic approach to (parameter) learning, starting from maximal likelihood estimation as a suitable natural transformation $\mathit{Flrn} \colon \mathcal{M}_* \Rightarrow \mathcal{D}$ from multisets to distributions. It then developed a more sophisticated form of data as multisets of predicates, with two associated forms, labeled as M/C, of validity and learning, even along a channel.

There are several avenues for further research.

- Extending the channel-based analysis from the E-part of the EM-algorithm to the whole of EM. This involves, in the context of the diagram (10), not only learning the state, but also learning the channel (as M-part). The Expectation-Maximisation algorithm involves iterating these E- and M-parts until some level of stability is reached. It will be described in an extended version of this paper.
- Extending this work to the more sophisticated form of learning called *Bayesian learning*, see [8, Ch.18], [16, §17.3] or [15, §6.1.2]. It is a form of higher order learning, where one does not immediately obtain the probability distribution in $\mathcal{D}(X)$, for a finite set X, but one obtains a *distribution over* $\mathcal{D}(X)$, typically in the form of Dirichlet distributions.

References

[1] Barber, D., "Bayesian Reasoning and Machine Learning," Cambridge Univ. Press, 2012, publicly available via http://web4.cs.ucl.ac.uk/staff/D.Barber/pmwiki/pmwiki.php?n=Brml.HomePage.

[2] Baum, L., T. Petrie, G. Soules and N. Weiss, *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains*, Ann. Math. Statistics **41** (1970), pp. 164–171.

[3] Bernardo, J. and A. Smith, "Bayesian Theory," John Wiley & Sons, 2000.

[4] Bishop, C., "Pattern Recognition and Machine Learning," Information Science and Statistics, Springer, 2006.

[5] Cho, K. and B. Jacobs, *The EfProb library for probabilistic calculations*, in: F. Bonchi and B. König, editors, *Conference on Algebra and Coalgebra in Computer Science (CALCO 2017)*, LIPIcs **72** (2017).

[6] Cho, K. and B. Jacobs, *Disintegration and Bayesian inversion, both abstractly and concretely* (2019), *Math. Struct. in Comp. Science*. See https://doi.org/10.1017/S0960129518000488 or arxiv.org/abs/1709.00322.

[7] Clerc, F., F. Dahlqvist, V. Danos and I. Garnier, *Pointless learning*, in: J. Esparza and A. Murawski, editors, *Foundations of Software Science and Computation Structures*, number 10203 in Lect. Notes Comp. Sci. (2017), pp. 355–369.

[8] Darwiche, A., "Modeling and Reasoning with Bayesian Networks," Cambridge Univ. Press, 2009.

[9] Dempster, A., N. Laird and D. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, Journ. Royal Statistical Soc. **39(1)** (1977), pp. 1–38.

[10] Do, C. and S. Batzoglou, *What is the expectation maximization algorithm?*, Nature Biotechnology **26** (2008), pp. 897–899.

[11] Jacobs, B., *From probability monads to commutative effectuses*, Journ. of Logical and Algebraic Methods in Programming **94** (2018), pp. 200–237.

[12] Jacobs, B. and F. Zanasi, *A predicate/state transformer semantics for Bayesian learning*, in: L. Birkedal, editor, *Math. Found. of Programming Semantics*, number 325 in Elect. Notes in Theor. Comp. Sci. (2016), pp. 185–200.

[13] Jacobs, B. and F. Zanasi, *A formal semantics of influence in Bayesian reasoning*, in: K. Larsen, H. Bodlaender and J.-F. Raskin, editors, *Math. Found. of Computer Science*, LIPIcs **83** (2017), pp. 21:1–21:14.

[14] Jacobs, B. and F. Zanasi, *The logical essentials of Bayesian reasoning*, in: *Probabilistic Programming* (book chapter, to appear in 2019), see arxiv.org/abs/1804.01193.

[15] Jensen, F. and T. Nielsen, "Bayesian Networks and Decision Graphs," Statistics for Engineering and Information Science, Springer, 2007, 2nd rev. edition.

[16] Koller, D. and N. Friedman, "Probabilistic Graphical Models. Principles and Techniques," MIT Press, Cambridge, MA, 2009.

[17] Pearl, J., "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," Graduate Texts in Mathematics 118, Morgan Kaufmann, 1988.

[18] Russell, S. and P. Norvig, "Artificial Intelligence. A Modern Approach," Prentice Hall, Englewood Cliffs, NJ, 2003.

A Appendix

We provide the missing proofs of Theorems 2.1 and 4.2 and of Proposition 4.3. The proof of the latter proposition is standard, but is included because it forms a proper preparation for the proofs of the two theorems — which are new results. All proofs rely on some basic real analysis for finding the maximum of functions with constraints on their inputs. This is done via the Lagrange multiplier method, see *e.g.* [4, §2.2]. This will be illustrated first. Subsequently we make use of a 'sum-increase' lemma to prove the other results.

Proof. (of Proposition 4.3) Let $\varphi \in \mathcal{M}(X)$ be a fixed non-empty multiset. We need to prove that the function $(-) \models_{\mathrm{M}} \varphi \colon \mathcal{D}(X) \to [0,1]$ takes its maximum at $\mathit{Flrn}(\varphi)$. We will thus seek the maximum of the function $\omega \mapsto \omega \models_{\mathrm{M}} \varphi$ by taking the derivative with respect to $\omega \in \mathcal{D}(X)$. We will work with the 'log-validity', that is, with the function $\omega \mapsto \ln(\omega \models_{\mathrm{M}} \varphi)$, where ln is the monotone (natural) logarithm

function. It reduces the product $\prod$ of powers in the definition of $\models_{\mathrm{M}}$ to a sum $\sum$ of multiplications.

Assume that the support of $\varphi = \sum_i r_i|x_i\rangle$ 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:

$$k(\boldsymbol{v}) \coloneqq \ln\Big(\textstyle\sum_i v_i|x_i\rangle \models_{\mathrm{M}} \sum_i r_i|x_i\rangle\Big) = \ln\Big(\prod_i v_i^{r_i}\Big) = \sum_i r_i \cdot \ln(v_i).$$

Since we have a constraint $(\sum_i v_i) - 1 = 0$ on the inputs, we can use the Lagrange multiplier method for finding the maximum. We thus take another parameter λ in a new function:

$$K(\boldsymbol{v}, \lambda) \coloneqq k(\boldsymbol{y}) - \lambda \cdot \big((\textstyle\sum_i v_i) - 1\big) = \big(\sum_i r_i \ln(v_i)\big) - \lambda \cdot \big((\textstyle\sum_i v_i) - 1\big).$$

The partial derivatives of K are:

$$\frac{\partial K}{\partial v_i}(\boldsymbol{v}, \lambda) = \frac{r_i}{v_i} - \lambda \qquad\qquad \frac{\partial K}{\partial \lambda}(\boldsymbol{v}, \lambda) = 1 - \textstyle\sum_i v_i.$$

Setting all of these to 0 and solving gives the required maximum. First, we have:

$$1 = \textstyle\sum_i v_i = \sum_i \frac{r_i}{\lambda} = \frac{\sum_i r_i}{\lambda}.$$

Hence $\lambda = \sum_i r_i$ and thus:

$$v_i = \frac{r_i}{\lambda} = \frac{r_i}{\sum_i r_i} \overset{(7)}{=} \mathit{Flrn}(\varphi)(x_i). \qquad \square$$

We now come to an auxiliary result which we shall call the sum-increase lemma. It is a special (discrete) case of a more general result [2, Thm. 2.1]. It describes how to find increases for sum expressions in general.

Lemma A.1 *Let X, Y be finite sets, and let $F\colon X \times Y \to \mathbb{R}_{\geq 0}$ be a given function. For each $x \in X$, write $F_1(x) \coloneqq \sum_{y \in Y} F(x, y)$. Assume that there is an $x' \in X$ with:*

$$x' = \operatorname*{argmax}_z G(x, z) \qquad \textit{where} \qquad G(x, z) \coloneqq \sum\nolimits_{y \in Y} F(x, y) \cdot \ln\big(F(z, y)\big).$$

Then $F_1(x') \geq F_1(x)$.

The proof uses Jensen's inequality: for $a_1, \ldots, a_n \in \mathbb{R}_{>0}$ and $r_1, \ldots, r_n \in [0, 1]$ with $\sum_i r_i = 1$ one has $\ln(\sum_i r_i a_i) \geq \sum_i r_i \ln(a_i)$. This gives a strict increase, except in 'corner' cases. The same holds for the above sum-increase lemma. The actual maximum x' in that lemma can in many situation be determined analytically — using the Lagrange multiplier method — but it need not be unique.

Proof. Let x' be the element where $G(x,-)\colon Y \to \mathbb{R}_{\geq 0}$ takes its maximum. This x' satisfies $F_1(x') \geq F_1(x)$, since:

$$\begin{aligned}
\ln\left(\frac{F_1(x')}{F_1(x)}\right) &= \ln\left(\sum_y \frac{F(x',y)}{F_1(x)}\right) = \ln\left(\sum_y \frac{F(x,y)}{F_1(x)} \cdot \frac{F(x',y)}{F(x,y)}\right) \\
&\geq \sum_y \frac{F(x,y)}{F_1(x)} \cdot \ln\left(\frac{F(x',y)}{F(x,y)}\right) \quad \text{by Jensen's inequality} \\
&= \frac{1}{F_1(x)} \cdot \sum_y F(x,y) \cdot \Big(\ln\big(F(x',y)\big) - \ln\big(F(x,y)\big)\Big) \\
&= \frac{1}{F_1(x)} \cdot \Big(G(x,x') - G(x,x)\Big) \;\geq\; 0. \qquad \square
\end{aligned}$$

Proof. (of Theorem 2.1) We have a state $\omega \in \mathcal{D}(X)$ and a predicate $p \in [0,1]^X$ and wish to prove a learning-style increase in validity: $\omega \models p \;\leq\; \omega|_p \models p$. We apply Lemma A.1 with $F\colon \mathcal{D}(X) \times X \to \mathbb{R}_{\geq 0}$ given by $F(\omega, x) = \omega(x) \cdot p(x)$. Then $F_1(\omega) = \sum_x F(\omega, x) = \sum_x \omega(x) \cdot p(x) = \omega \models p$.

In the notation of Lemma A.1 we have $G(\omega, \omega') = \sum_x \omega(x) \cdot p(x) \cdot \ln\big(\omega'(x) \cdot p(x)\big)$ and we wish to find the maximum of $G(\omega, -)\colon \mathcal{D}(X) \to \mathbb{R}_{\geq 0}$. Let $X = \{x_1, \ldots, x_n\}$. We can see $G(\omega, -)$ as a function $\mathbb{R}^n \to \mathbb{R}_{\geq 0}$ with constraints on its inputs.

These constraints are handled, as before, via the Lagrange multiplier method for finding the maximum. We keep ω fixed and consider a new function H with an additional parameter λ.

$$\begin{aligned}
H(\boldsymbol{v}, \lambda) &\coloneqq G(\omega, \boldsymbol{v}) - \lambda \cdot \big((\textstyle\sum_i v_i) - 1\big) \\
&= \textstyle\sum_i \omega(x_i) \cdot p(x_i) \cdot \ln\big(v_i \cdot p(x_i)\big) - \lambda \cdot \big((\textstyle\sum_i v_i) - 1\big).
\end{aligned}$$

The numbers v_i are variables for the unknowns $\omega'(x_i)$.

The partial derivatives of H are:

$$\frac{\partial H}{\partial v_i}(\boldsymbol{v}, \lambda) = \frac{\omega(x_i) \cdot p(x_i)}{v_i \cdot p(x_i)} \cdot p(x_i) - \lambda = \frac{\omega(x_i) \cdot p(x_i)}{v_i} - \lambda \qquad \frac{\partial H}{\partial \lambda}(\boldsymbol{v}, \lambda) = 1 - \textstyle\sum_i v_i.$$

Setting all these derivatives to zero yields:

$$1 = \textstyle\sum_i v_i = \sum_i \dfrac{\omega(x_i) \cdot p(x_i)}{\lambda} = \dfrac{\omega \models p}{\lambda}.$$

Hence $\lambda = \omega \models p$ and thus:

$$v_i = \frac{\omega(x_i) \cdot p(x_i)}{\lambda} = \frac{\omega(x_i) \cdot p(x_i)}{\omega \models p} = \omega|_p(x_i). \qquad \square$$

Proof. (of Theorem 4.2 (i)) Let $\omega \in \mathcal{D}(X)$ be state on a finite set X and let $p_1, \ldots, p_n$ be predicates on X, all with non-zero validity $\omega \models p_i$. We claim that the state $\omega' = \sum_i \frac{1}{n} \cdot \omega|_{p_i}$ then satisfies:

$$\prod\nolimits_i (\omega' \models p_i) \;\geq\; \prod\nolimits_i (\omega \models p_i). \tag{A.1}$$

The inequality in Theorem 4.2 (i) is a direct consequence of (A.1). We shall prove (A.1) for $n = 2$. The generalisation to arbitrary n should then be obvious, but involves much more book-keeping of additional variables.

We use Lemma A.1 with function $F \colon \mathcal{D}(X) \times X \times X \to \mathbb{R}_{\geq 0}$ given by:

$$F(\omega, x, y) \coloneqq \omega(x) \cdot p_1(x) \cdot \omega(y) \cdot p_2(y).$$

Then by distributivity of multiplication over addition:

$$\textstyle\sum_{x,y} F(\omega, x, y) = \big(\sum_x \omega(x) \cdot p_1(x)\big) \cdot \big(\sum_y \omega(y) \cdot p_2(y)\big) = (\omega \models p_1) \cdot (\omega \models p_2).$$

Let $X = \{x_1, \ldots, x_n\}$ and let the function H be given by:

$$H(\boldsymbol{v}, \lambda) \coloneqq \textstyle\sum_{i,j} F(\omega, x_i, x_j) \cdot \ln\big(v_i \cdot p_1(x_i) \cdot v_j \cdot p_2(x_j)\big) \;-\; \lambda \cdot \big((\sum_i v_i) - 1\big).$$

Then:

$$\frac{\partial H}{\partial v_k}(\boldsymbol{v}, \lambda) = \sum_i \frac{F(\omega, x_k, x_i) + F(\omega, x_i, x_k)}{v_k} \;-\; \lambda \qquad \frac{\partial H}{\partial \lambda}(\boldsymbol{v}, \lambda) = 1 - \textstyle\sum_i v_i.$$

Setting these to zero gives:

$$1 = \textstyle\sum_k v_k = \dfrac{\sum_{k,i} F(\omega, x_k, x_i) + F(\omega, x_i, x_k)}{\lambda} = \dfrac{2 \cdot (\omega \models p_1) \cdot (\omega \models p_2)}{\lambda}.$$

Hence $\lambda = 2 \cdot (\omega \models p_1) \cdot (\omega \models p_2)$ so that:

$$\begin{aligned} v_k &= \frac{\sum_i F(\omega, x_k, x_i) + F(\omega, x_i, x_k)}{\lambda} \\ &= \tfrac{1}{2} \cdot \frac{\omega(x_k) \cdot p_1(x_k) \cdot (\omega \models p_2)}{(\omega \models p_1) \cdot (\omega \models p_2)} + \tfrac{1}{2} \cdot \frac{(\omega \models p_1) \cdot \omega(x_k) \cdot p_2(x_k)}{(\omega \models p_1) \cdot (\omega \models p_2)} \\ &= \tfrac{1}{2} \cdot \frac{\omega(x_k) \cdot p_1(x_k)}{\omega \models p_1} + \tfrac{1}{2} \cdot \frac{\omega(x_k) \cdot p_2(x_k)}{\omega \models p_2} \\ &= \tfrac{1}{2} \cdot \omega|_{p_1}(x_k) + \tfrac{1}{2} \cdot \omega|_{p_2}(x_k). \end{aligned}$$

□