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. Three different forms of such learning are distinguished, in terms of additive and multiplicative predicate transformation along a channel, and in terms of a dagger of a channel. Two of these three forms are illustrated with conflicting examples from the literature, which both claim to form the first half of the famous Expectation-Maximisation algorithm.

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 [1,2,8,14,15,16]. 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 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], [15, §17.1] or [14, §6.1.1].

Often one wishes to learn 'latent' or 'hidden' variables on a probability space X, which is only indirectly accessible. In the current setting this means that we have (multiset) data, not on X itself, but on a different space Y, with a channel (Kleisli map) $X \mathrel{-\!\!\circ\!\!\to} Y$ between them. In Section 6 we distinguish three forms of learning along such a channel, two of which are called additive and multiplicative. They arise in a systematic manner via a free extension using additive and multiplicative cone structures on predicates. The third form arises via a dagger of a channel, as studied in [7,6]. The differences between these three forms of learning are illustrated via (parts of) examples from the literature, involving the famous Expectation-Maximisation (EM)

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

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

algorithm that is often used for probabilistic data classification. The novel channel-based formalisation in this paper uncovers that there are different interpretations of what (the E-part of) EM means. This somewhat uncomfortable fact suggests that a deeper analysis of such probabilistic classification is needed.

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:

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

where $\mathit{supp}(\phi)$ is the support of ϕ, *i.e.* the subset $\{x \in X \mid \phi(x) \neq 0\}$. 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:

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

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 $(g \circ f)(x) = g \gg f(x)$.

We need the following result about cones, whose third point is maybe less standard. We recall that a cone is very much like a vector space, except that the scalars — in this case $\mathbb{R}_{\geq 0}$ — are not a field but only a semiring.

Proposition 2.1 *For each set X,*

(i) *the sets $(\mathbb{R}_{\geq 0})^X$ and $\mathcal{M}(X)$ are* additive cones*, via pointwise addition $(+, \mathbf{0})$ of multiplicities and via pointwise multiplication of multiplicities with a scalar from $\mathbb{R}_{\geq 0}$;*

(ii) *in fact, $\mathcal{M}(X)$ is the* free *cone on the set X;*

(iii) *the sets $(\mathbb{R}_{\geq 0})^X$ and $[0,1]^X$ are also* multiplicative cones*, via pointwise multiplication $(\&, \mathbf{1})$ and via pointwise power with a scalar from $\mathbb{R}_{\geq 0}$; $\mathcal{M}(X)$ is a multiplicative cone if X is a finite set.* □

The finiteness requirement is needed in the last point to make sure that the constant-1 function $\mathbf{1} \colon X \to \mathbb{R}_{\geq 0}$ is a multiset. The addition, multiplication and two scalar operations involved in the above proposition are:

$$\begin{cases} (\phi + \psi)(x) = \phi(x) + \psi(x) \\ (r \cdot \phi)(x) = r \cdot \phi(x) \end{cases} \qquad \begin{cases} (\phi \mathbin{\&} \psi)(x) = \phi(x) \cdot \psi(x) \\ (\phi^r)(x) = \phi(x)^r. \end{cases}$$

We often refer to functions in $[0,1]^X$ and in $(\mathbb{R}_{\geq 0})^X$ as *predicates* on X.

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 (1) 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

(1)

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:

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

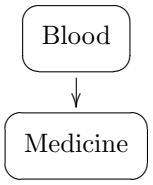
We can normalise this multiset τ. It yields a joint probability distribution:

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

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

(2) The first and second marginals $\mathsf{M}_1(\omega)$ and $\mathsf{M}_2(\omega)$ 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 (1), via the normalisation of 'totals' column and row. This fact looks like a triviality, but involves a naturality property (see Lemma 4.2 below).

(3) Next we wish to use the above table (1) 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 1, 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.* [12,13]) 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 [7,6]. 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 (1). This is a highly relevant property, which essentially means that (this kind of) learning can be done locally.

This 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 analysed from a systematic categorical perspective.

4 Frequentist learning by counting

As mentioned in the introduction, maximal likelihood estimation (MLE) is one kind of parameter learning, see *e.g.* [15,8,14]. We reframe it here as frequentist learning. 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 4.1 *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 $\ell_X\colon \mathcal{M}_*(X) \to \mathcal{D}(X)$, determined by:*

$$\ell_X(\phi)(x) \coloneqq \frac{\phi(x)}{\sum_y \phi(y)} \qquad \textit{i.e.} \qquad \ell_X\big(\textstyle\sum_i r_i|x_i\rangle\big) = \displaystyle\sum_i \frac{r_i}{\sum_j r_j}|x_i\rangle \tag{2}$$

Lemma 4.2 *The maps $\ell_X\colon \mathcal{M}_*(X) \to \mathcal{D}(X)$ 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 $h\colon X \to Y$,

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

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 ℓ 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 how a likelihood function arises, which is used to show that $\ell(\phi)$ is optimal, in a suitable sense.

Consider the evaluation mapping $ev\colon X \to [0,1]^{\mathcal{D}(X)}$, given by $ev(x)(\omega) = \omega(x)$. We can freely extend this map to $\overline{ev}\colon \mathcal{M}(X) \to [0,1]^{\mathcal{D}(X)}$, using the multiplicative structure on predicates, see Proposition 2.1 (ii), (iii). Then:

$$\overline{ev}(\phi) \;=\; \&_x\, ev(x)^{\phi(x)} \qquad \text{so} \qquad \overline{ev}(\phi)(\omega) \;=\; \textstyle\prod_x \omega(x)^{\phi(x)}. \tag{3}$$

The next result describes a fundamental property, see *e.g.* [15, Ex. 17.5]. It explains the name 'maximal likelihood estimation' for the maps ℓ.

Proposition 4.3 *For a multiset $\phi \in \mathcal{M}_*(X)$ the likelihood predicate:*

$$\mathcal{D}(X) \xrightarrow{\overline{ev}(\phi)} [0,1]$$

takes its maximum at the learned distribution $\ell_X(\phi) \in \mathcal{D}(X)$. □

The proof uses the Lagrange multiplier method, see *e.g.* [3, §2.2].

5 Learning by conditioning

In this section we show how the learning map ℓ from the previous section can be expressed via conditioning of a uniform state. This will allow us to express to a (multiplicative) compositionality result.

But first we need some basic definitions, see *e.g.* [10,13] for more information. Let $\omega \in \mathcal{D}(X)$ be state and $p \in (\mathbb{R}_{\geq 0})^X$ be predicate, both on the same set X. We define the *validity* $\omega \models p$ of predicate p in state ω, or *expected value* as the (finite) sum:

$$\omega \models p \;:=\; \textstyle\sum_x \omega(x)\cdot p(x).$$

If this validity is non-zero, we can define the updated state $\omega|_p \in \mathcal{D}(X)$ via conditioning as:

$$\omega|_p(x) \;:=\; \frac{\omega(x)\cdot p(x)}{\omega \models p}. \tag{4}$$

It is not hard to see that multiple conditionings can be reduced to a single conditioning via multiplication of predicates:

$$\big(\omega|_p\big)|_q \;=\; \omega|_{p\&q} \;=\; \big(\omega|_q\big)|_p. \tag{5}$$

As an aside: since the order of updating does not matter, it is appropriate to use *multisets* of data, in which only the multiplicities of elements matter, but not their order.

We can now express learning as conditioning of a uniform state.

Proposition 5.1 *Let $\phi \in \mathcal{M}_*(X) \subseteq (\mathbb{R}_{\geq 0})^X$ be multiset with (finite) support $S = supp(\phi) \subseteq X$. Then:*

(i) $\ell(\phi) = \upsilon|_\phi$, *where υ is the uniform distribution on S;*
(ii) $\ell(r\cdot\phi) = \ell(\phi)$, *for $r \in \mathbb{R}_{>0}$;*
(iii) $\ell(\phi \,\&\, \psi) = \ell(\phi)|_\psi$.

Proof. (i) Let the support S of φ have n elements, so that $\upsilon = \sum_{x \in S} \frac{1}{n}|x\rangle$. We simply follow the definition of conditioning (4):

$$\upsilon|_\phi(x) = \frac{\upsilon(x) \cdot \phi(x)}{\upsilon \models \phi} = \frac{1/n \cdot \phi(x)}{\sum_{y \in S} 1/n \cdot \phi(y)} = \frac{\phi(x)}{\sum_{y \in S} \phi(y)} = \ell(\phi)(x).$$

(ii) Obvious.

(iii) We use point (i) twice and (5) in:

$$\ell(\phi \,\&\, \psi) = \upsilon|_{\phi \& \psi} = \upsilon|_\phi|_\psi = \ell(\phi)|_\psi. \qquad \square$$

Now that we have seen the first point (i) we can also define learning from data ϕ with a prior distribution ω as conditioning $\omega|_\phi$. The last point (iii) 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 $\ell(\phi + \psi)$, since then we can (later) add new data to what we have already learnt from an existing table (multiset) of data.

6 Learning along a channel

In frequentist learning we have a multiset of data on a set X and turn it in an associated 'empirical' distribution on the same X; this distribution gives the highest likelihood to the data, see Proposition 4.3. In this section we consider the slightly more complicated situation where we have a known channel $e\colon X \multimap Y$ and a multiset of data on the codomain Y of the channel. We now like to learn the associated distribution on the domain X. This problem is often described in terms of 'hidden' or 'latent' variables, since the elements of the domain X are not directly accessible, only indirectly via the 'emission' channel e.

We shall distinguish three approaches for learning along a channel, via in particular the additive and multiplicative cone structures of Proposition 2.1, and in addition a backtracking technique. These three approaches are not clearly distinguished in the literature, as will be illustrated in several examples.

6.1 Additive learning along a channel

In order to describe the additive learning approach we first need to describe what is commonly called *predicate transformation* along a channel $e\colon X \multimap Y$. It turns a predicate q on the codomain Y into a predicate $e \ll q$ on the domain X, via the standard definition $(e \ll q)(x) = \sum_y q(y) \cdot e(x)(y)$. Here we shall redescribe this definition as a free extension, using the *additive* cone structure on predicates on X. This works as follows. First we rearrange the channel as a 'point transformation' function $pt_e\colon Y \to [0,1]^X$, namely $pt_e(y)(x) \coloneqq e(x)(y) = (e \ll \mathbf{1}_y)(x)$. Here we write $\mathbf{1}_y\colon Y \to [0,1]$ for the 'Dirac' predicate, which is 1 at y and 0 everywhere else. We then form its unique extension to multisets $\overline{pt_e}\colon \mathcal{M}(Y) \to [0,1]^X$, given by:

$$\begin{array}{rcll} \overline{pt_e}(\psi) & = & \sum_y \psi(y) \cdot pt_e(y), & \text{that is,} \\ \overline{pt_e}\big(\sum_j s_j|y_j\rangle\big)(x) & = & \sum_j s_j \cdot e(x)(y_j) & = \Big(e \ll \big(\sum_j s_j|y_j\rangle\big)\Big)(x). \end{array} \tag{6}$$

Definition 6.1 *Let $e\colon X \multimap Y$ is a channel with data $\psi \in \mathcal{M}_*(Y)$ on its codomain Y.* Additive learning *$a\ell$ of ψ along the channel e is defined as frequentist learning ℓ with the (additively) transformed data:*

$$a\ell(\psi, e) \coloneqq \ell(e \ll \psi) = \upsilon|_{e \ll \psi}. \tag{7}$$

The latter equation comes from Proposition 5.1 (i), and allows us to do additive learning with a prior ω, which then replaces the uniform distribution υ.

Explicitly, Equation (7) becomes, for an element $x \in X$,

$$a\ell\big(\textstyle\sum_j s_j|y_j\rangle, e\big)(x) = \dfrac{\sum_j s_j \cdot e(x)(y_j)}{\sum_j s_j \cdot \sum_z e(z)(y_j)}. \tag{8}$$

We shall postpone an illustration to the next version of learning. At this stage we just like to remark that additive learning $a\ell(\phi, \mathrm{id})$ along the identity channel is simply ordinary learning $\ell(\phi)$. Moreover, additive

learning interacts appropriately with channel composition, as in:

$$a\ell(\chi, d \mathbin{\odot} c) \;=\; \upsilon|_{(d \odot c) \ll \chi} \;=\; \upsilon|_{c \ll (d \ll \chi)} \;=\; a\ell(d \ll \chi, c).$$

6.2 Multiplicative learning along a channel

Multiplicative learning along a channel $e\colon X \nrightarrow Y$ starts from the same point transformation map $pt_e\colon Y \to [0,1]^X$ as for additive learning. The difference is that we now use the *multiplicative* cone structure on predicates $[0,1]^X$, see Proposition 2.1 (iii). We shall write the resulting unique extension with double overlines, to distinguish it from the single overline version (6). We then get $\overline{\overline{pt_e}}\colon \mathcal{M}(Y) \to [0,1]^X$ in:

$$e \lll \psi \;:=\; \overline{\overline{pt_e}}(\psi) \;=\; \&_y \; pt_e(y)^{\psi(y)}. \tag{9}$$

This new 'multiplicative' predicate transformation $\lll$ can be described explicitly as:

$$\Big(e \lll \big(\textstyle\sum_j s_j |y_j\rangle\big)\Big)(x) \;=\; \textstyle\prod_j e(x)(y_j)^{s_j}. \tag{10}$$

The definition of multiplicative learning is as for additive learning, except that (additive) predicate transformation $\ll$ is replaced by multiplicative predicate transformation $\lll$.

Definition 6.2 *For a channel $e\colon X \nrightarrow Y$ and a multiset $\psi \in \mathcal{M}_*(Y)$ we define* multiplicative learning *$m\ell$ of ψ along e as frequentist learning ℓ with the multiplicatively transformed data:*

$$m\ell(\psi, e) \;:=\; \ell(e \lll \psi) \;=\; \upsilon|_{e \lll \psi}. \tag{11}$$

Explicitly, for $x \in X$,

$$m\ell\big(\textstyle\sum_j s_j |y_j\rangle, e\big)(x) \;=\; \dfrac{\prod_j e(x)(y_j)^{s_j}}{\sum_z \prod_j e(z)(y_j)^{s_j}}. \tag{12}$$

In the remainder of this subsection we consider an example from [9]. Consider five separate data sets (multisets), each with ten coin outcomes head (H) and tail (T), see the first column in the table (13) 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 additive and multiplicative learning along the coin channel e, and also already for backtrack learning — to be discussed in the next subsection. Each line describes a new learning action, independent from the outcomes of earlier lines.

data ϕ	**additive** $a\ell(\phi, e)$	**multiplicative** $m\ell(\phi, e)$	**backtrack** $b\ell(\phi, e)$
$5\|H\rangle + 5\|T\rangle$	$0.5\|0\rangle + 0.5\|1\rangle$	$0.4491\|0\rangle + 0.5509\|1\rangle$	$0.4949\|0\rangle + 0.5051\|1\rangle$
$9\|H\rangle + 1\|T\rangle$	$0.537\|0\rangle + 0.463\|1\rangle$	$0.805\|0\rangle + 0.195\|1\rangle$	$0.5354\|0\rangle + 0.4646\|1\rangle$
$8\|H\rangle + 2\|T\rangle$	$0.5283\|0\rangle + 0.4717\|1\rangle$	$0.7335\|0\rangle + 0.2665\|1\rangle$	$0.5253\|0\rangle + 0.4747\|1\rangle$
$4\|H\rangle + 6\|T\rangle$	$0.4898\|0\rangle + 0.5102\|1\rangle$	$0.3522\|0\rangle + 0.6478\|1\rangle$	$0.4848\|0\rangle + 0.5152\|1\rangle$
$7\|H\rangle + 3\|T\rangle$	$0.5192\|0\rangle + 0.4808\|1\rangle$	$0.6472\|0\rangle + 0.3528\|1\rangle$	$0.5152\|0\rangle + 0.4848\|1\rangle$

(13)

This table is obtained via the formulas (8) and (12) and (16). We mostly ignore the last column for a moment. We see that the multiplicative approach $m\ell$ is best at picking up the differences between the numbers of heads and tails in the data. This third $m\ell$ column is exactly as reported in [9], although the term 'multiplicative learning' does not occur there; the computations are obtained in [9] via an explictly given formula that coincides with (12) above. The discriminating power of multiplicative learning may be the reason why it is frequently

used — especially in the Expectation-Maximisation (EM) algorithm. This table concentrates only on the 'E' part of this 'EM' algorithm.

The following result gives another advantage of multiplicative learning: it is additively compositional, so that successive data can be incorporated.

Proposition 6.3 *Multiplicative learning $m\ell$ along a channel e satisfies:*

$$m\ell(\phi + \psi, e) \;=\; m\ell(\phi, e)|_{e \lll \psi}.$$

Proof. The crucial point is that multiplicative predicate transformation sends sums to products: $e \lll (\phi + \psi) = (e \lll \phi) \,\&\, (e \lll \psi)$. This holds almost by definition, but still we give an explicit proof:

$$\begin{array}{rcl}
\Big(e \lll (\phi + \psi)\Big)(x) & \overset{(10)}{=} & \prod_y e(x)(y)^{\phi(y)+\psi(y)} \\
& = & \prod_y e(x)(y)^{\phi(y)} \cdot e(x)(y)^{\psi(y)} \\
& = & \prod_y e(x)(y)^{\phi(y)} \cdot \prod_y e(x)(y)^{\psi(y)} \\
& \overset{(10)}{=} & \Big(e \lll \phi\Big)(x) \cdot \Big(e \lll \psi\Big)(x) \\
& = & \Big(\big(e \lll \phi\big) \,\&\, \big(e \lll \psi\big)\Big)(x).
\end{array}$$

Now we are almost done:

$$\begin{array}{rcll}
m\ell(\phi + \psi, e) \overset{(11)}{=} \upsilon|_{e \lll (\phi+\psi)} & = & \upsilon|_{(e \lll \phi)\&(e \lll \psi)} & \text{as just shown} \\
& = & \upsilon|_{e \lll \phi}|_{e \lll \psi} & \text{by (5)} \\
& \overset{(11)}{=} & m\ell(\phi, e)|_{e \lll \psi}. &
\end{array}$$

□

A (mathematical) disadvantage of multiplicative learning is that it does not interact well with identity channels and channel composition.

6.3 Backtrack learning along a channel

The third form of learning along a channel involves the so-called 'dagger' or 'Bayesian inversion' of that channel, see [7,6]. 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 [7,6] and the references given there for more information. Here we use it in our third version of learning along a channel.

Definition 6.4 *For a channel $e \colon X \nrightarrow Y$ and a multiset $\psi \in \mathcal{M}_*(Y)$ we define* backtrack learning *$b\ell$ of ψ along e as state transformation with the inverted dagger channel, applied the state obtained by frequentist learning ℓ from the data:*

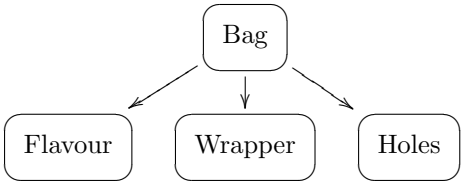
$$b\ell(\psi, e) \coloneqq e^\dagger_\upsilon \gg \ell(\psi). \tag{15}$$

This means that for $x \in X$,

$$b\ell\big(\textstyle\sum_j s_j |y_j\rangle, e\big)(x) = \sum_j \dfrac{s_j \cdot e(x)(y_j)}{s \cdot \sum_z e(z)(y_j)} \quad \text{where } s = \textstyle\sum_j s_j. \tag{16}$$

The additive and backtrack forms of learning along a channel, $a\ell$ and $b\ell$, bear similarities to the different forms of updating along a channel, associated with Pearl and Jeffrey, respectively, see [4,11]. Jeffrey's version involves backtracking, in the form of adjustment to a new situation, and is also formalised via a dagger channel, see [11] for details. The term 'backtracking' is copied from that setting. In Table (13) we have already seen that backtrack learning produces different outcomes from additive and multiplicative learning.

We conclude with an example from [17, §20.3], as part of its illustration of the EM-algorithm. It involves the Bayesian network on the right, 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 [17]:



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

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:

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

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

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

We can now calculate backtrack learning from these data via the dagger of the tuple channel. In [17] this is not done from the uniform distribution υ, like in (15), but from a prior distribution $\rho = \frac{6}{10}|0\rangle + \frac{4}{10}|1\rangle$. We then compute the learned distribution on $\{0,1\}$ as backtrack outcome:

$$\langle f, w, h\rangle^\dagger_\rho \gg \ell(\psi) \;=\; 0.6124|0\rangle + 0.3876|1\rangle.$$

This probability 0.6124 is exactly as computed in [17, §20.3], but without the channel machinery.

We thus notice that [9] and [17] perform completely different computations in their explanation of Expectation-Maximisation, using, respectively, in the terminology of our setting, multiplicative and backtrack learning along a channel. It is unclear why they do different things.

7 Conclusions and further work

This paper has developed a systematic approach to parameter learning, using universal properties, different cone structures, and daggers. It has first described maximal likelihood estimation as a suitable natural transformation $\ell\colon \mathcal{M}_* \Rightarrow \mathcal{D}$ and used it subsequently in three different forms of learning along a channel. Since learning with ℓ yields a maximal distribution, see Proposition 4.3, these three forms of learning along a channel can also be described via maximality.

There are several avenues for further research.

- Getting a better understanding of the differences between the three forms of learning along a channel and , in particular, of the circumstances when to use which form.
- Extending the channel-based analysis from the E-part of the EM-algorithm to the whole of EM.
- Extending this work to the more sophisticated form of learning called *Bayesian learning*, see [8, Ch.18], [15, §17.3] or [14, §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* $\mathcal{D}(X)$, 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] Bernardo, J. and A. Smith, "Bayesian Theory," John Wiley & Sons, 2000.

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

[4] Chan, H. and A. Darwiche, *On the revision of probabilistic beliefs using uncertain evidence*, Artif. Intelligence **163** (2005), pp. 67–90.

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

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

[11] Jacobs, B., *A mathematical account of soft evidence, and of Jeffrey's 'destructive' versus Pearl's 'constructive' updating* (2018), see arxiv.org/abs/1807.05609.

[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, *The logical essentials of Bayesian reasoning*, in: *Probabilistic Programming* (book chapter, to appear in 2019), see arxiv.org/abs/1804.01193.

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

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

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

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