A predicate/state transformer semantics for Bayesian learning

Bart Jacobs and Fabio Zanasi

Radboud University Nijmegen, The Netherlands

Abstract

This paper establishes a link between Bayesian inference (learning) and predicate and state transformer operations from programming semantics and logic. Specifically, a very general definition of backward inference is given via first applying a predicate transformer and then conditioning. Analogously, forward inference involves first conditioning and then applying a state transformer. These definitions are illustrated in many examples in discrete and continuous probability theory and also in quantum theory.

Keywords: Inference, learning, Bayes, Kleisli category, effectus, predicate transformer, state transformer

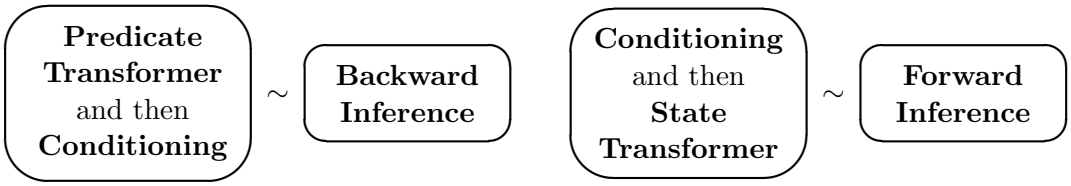
1 Introduction

Increasingly probabilistic programs are used to describe problems in Bayesian inference ([2]), see *e.g.* [10,19,4,1,21]. The term 'inference' is used for what is informally best called: learning [1]. Learning involves updating one's knowledge, in the light of certain evidence, typically given via the validity of a certain predicate (which may be a fuzzy one). In this situation one represents knowledge in terms of likelihoods, via a probability distribution (in the discrete case) or a probability measure (in the continuous case). Updating one's knowledge then involves computing a conditional distribution/measure.

Now that the overlap between the (probabilistic) programming community and the Bayesian community is growing, a merging of concepts and techniques can be expected. This paper is an example. It shows how the notions of predicate and state transformer from programming languages semantics ([7]) can be used in precisely defining two fundamental notions of learning: backward and forward inference. A conditioning operation, which makes a certain distribution/measure depend on a predicate, also plays a role. In a nutshell, the correspondence can be summarised

[1] The Bayesian community associates learning to various tasks. A prominent learning task is finding a way of finding out what the topology of a network is, based on (in)dependence relations, starting from a big joint distribution.

as follows.



This connection hopefully works as an *Aha Erlebnis*, giving a sudden insight. Indeed, predicate transformers work backwards, from predicates on post-states to predicates on pre-states. This is precisely what is at stake in backward inference — as we will demonstrate. Similarly, state transformers work in a forward direction, which is what happens in forward learning.

Strictly speaking, the main contribution of this paper is only one definition, namely of (backward and forward) inference, see Definition 2.1. Contrarily to traditional approaches, our formulation is not tied to the probabilistic setting, but works in the context of any *effectus*, that is a categorical notion embracing a wide spectrum of computational models, both classical, probabilistic and also quantum, see [11,5]. Within the theory of effectuses, predicate and state transformers are well-defined, and predicates (or effects) and states can be nicely organised in state-and-effect triangles, which connect predicates and states via a (dual) adjunction (1), see also [12]. Intriguingly, these triangles correspond to what physicists call the duality between states and effects, referring to the opposite directions in the work of Schrödinger and Heisenberg on quantum foundations. Within this effectus context one can also describe normalisation and conditioning of states in an abstract manner (see [13,5]). Therefore, we believe effectuses form the right setting for developing a general approach to inference.

Still, precisely recognising what is what in this setting is a subtle matter. For instance, what is a predicate, at the abstract level? Traditionally in probability theory 'events' are used as predicates. Formally they are subsets of the sample space, corresponding to 'sharp' predicates on this space. More generally, 'fuzzy' predicates are considered; they are functions taking values in the unit interval $[0, 1]$. The sharp predicates can then be characterised as the ones taking values in the Boolean subset $\{0, 1\} \subseteq [0, 1]$. In discrete probability every distribution is at the same time a fuzzy predicate. This blurs the picture — the confusion between states and predicates is particularly evident in Bayesian network representations, where nodes may play both roles. In continuous probability there is, in principle, a clear distinction between states (probability measures) and predicates (measurable functions to $[0, 1]$). But again, things easily get mixed up, when a state/measure is given by a probability distribution function (pdf), which looks very much like a predicate. The framework of effectus theory helps in this respect, since it gives a clear distinction between states, as maps of the form $1 \to X$, and predicates, as maps $X \to 1+1$. Only when this perspective is recognised, the role of predicate and state transformers becomes clear. It is for this reason that we think it is justified to dedicate an entire paper to elaborating and explaining a single definition.

The paper is organised as follows. We first introduce the notions of backward and forward inference in terms of predicate and state transformers and show some

basic properties. Then, we concentrate on illustrating the impact and power of our definition in many situations. We show what our abstract setting translates to in discrete and continuous probability theory and also (briefly) in quantum theory. We elaborate many examples of computations of how inference works, and what it produces. Of special interest is the application of our definition of inference in Bayesian networks. It is shown that the forward/backward distinction can be used flexibly, and can describe what inference means at different points in the network.

2 Backward and forward inference, abstractly

In this section we describe our abstract set up for inference, both in a backward and forward manner. This works in the setting of an *effectus*: briefly, this is a category with finite coproducts $(+, 0)$ and a final object 1, such that certain diagrams are pullbacks and certain maps are jointly monic. By virtue of these basic requirements, an effectus is able to capture some basic aspects of quantum computation, with probabilistic computation as special case, see [11,5].

States in an effectus $\mathbf{C}$ are maps of the form $1 \to X$ and *predicates* are maps $X \to 2 = 1 + 1$. The set of states $\mathrm{Stat}(X)$ of an object X form a convex set, and the set of predicates $\mathrm{Pred}(X)$ on X form an effect module. States and predicates give rise to a 'state-and-effect triangle' of the form:

$$\begin{array}{ccccc} \mathbf{EMod}^{\mathrm{op}} & & \top & & \mathbf{Conv} \\ & {\scriptstyle \mathrm{Pred}=\mathrm{Hom}(-,2)} \searrow & & \swarrow {\scriptstyle \mathrm{Stat}=\mathrm{Hom}(1,-)} & \\ & & \mathbf{C} & & \end{array} \tag{1}$$

We refer to [11] for details about effect modules and convex sets. In the current setting we need the *predicate transformer* $f^* = \mathrm{Pred}(f)$ and *state transformer* $f_* = \mathrm{Stat}(f)$ operations associated with a map $f \colon X \to Y$ in the base category $\mathbf{C}$. They are given by pre- and post-composition:

$$\mathrm{Pred}(X) \xleftarrow{f^* = (-) \circ f} \mathrm{Pred}(Y) \qquad\qquad \mathrm{Stat}(X) \xrightarrow{f_* = f \circ (-)} \mathrm{Stat}(Y)$$

In concrete examples of effectuses states are distributions — in the Kleisli category of the distribution monad — or probability measures — in the Kleisli category of the Giry monad — or just states — in C^*- or W^*-algebras. We will understand states as descriptions of our state of knowledge. Given a predicate p and a state ω on the same object X two definitions are of interest:

$$\omega \models p := p \circ \omega \qquad \text{and} \qquad \omega|_p, \ \text{ the conditional state on } X. \tag{2}$$

The expression $\omega \models p$ describes the validity, or expected value, of the predicate in the state ω. Typically its value is in the unit interval $[0, 1]$. If this validity $\omega \models p$ is non-zero, then the conditional state $\omega|_p$ exists. It is the updated state of knowledge after observing 'evidence' p. In each of the above concrete examples of states we can define such conditional states (see below). In fact, conditioning can be defined abstractly in the theory of effectuses, using 'assert' maps, see [5, Example 58], but we don't need such a level of abstraction here.

We now distinguish two forms of inference (learning).

Definition 2.1 *Backward inference* $\omega|_{f^*(p)}$ involves first applying a predicate transformer and then computing a conditional. This applies in situations of the form:

$$1 \xrightarrow{\ \omega\ } X \xrightarrow{\ f\ } Y \xrightarrow{\ q\ } 1+1 \tag{3}$$

More explicitly, one first applies the predicate transformer f^* to the predicate q on Y, and then computes the backwardly inferred conditional state $\omega|_{f^*(q)}$ on X.

Forward inference $f_*(\omega|_p)$ is first computing a conditional and then applying a state transformer. This works in a situation:

$$\begin{array}{ccccc} 1 & \xrightarrow{\ \omega\ } & X & \xrightarrow{\ f\ } & Y \\ & & \downarrow{\scriptstyle p} & & \\ & & 1+1 & & \end{array} \tag{4}$$

In this case the conditional state on X is $\omega|_p$, and applying the state transformer f_* gives the forwardly inferred state $f_*(\omega|_p)$ on Y.

In the trivial case where the map f is the identity there is no difference between backward and forward inference. Inference then just involves updating a state (of knowledge). Notice that in backward inference we use a predicate on the codomain of the map f, namely q, and update our knowledge about the state on f's domain X. In forward inference we use a predicate on the domain of f, namely p, and use it to infer more about the state on f's codomain Y. This may also be called 'evidence propagation'.

In the situation (4) we have the following Galois style equalities for validity:

$$\big(\omega \models f^*(q)\big) = q \circ f \circ \omega = \big(f_*(\omega) \models q\big).$$

In general, there are very few 'nice' algebraic properties for conditional states. For instance, we do have $f_*\big(\omega|_{f^*(q)}\big) = \big(f \circ \omega\big)|_q$, but only for the special case where the map f is 'pure'. The latter means for instance in a Kleisli category that the map comes from the underlying category.

In the remainder of this paper we shall illustrate these forms of inference via several examples, involving various kinds of computation, and including Bayesian networks where the above map f in (3) and (4) arises from a graph (network of conditional probability tables). The composition notation '$\circ$' used above looks deceptively simple, but will each time be interpreted differently in different categories. This leads to various concrete forms of inference which are all instances of the same pattern.

3 Inference with discrete probability

We shall write $\mathcal{D}$ for the discrete probability monad on the category **Set** of sets and functions. The set $\mathcal{D}(X)$ contains the finite discrete probability distributions

ω over X which we write as formal convex combinations:

$$\omega = r_1 \,|x_1\rangle + \cdots + r_n \,|x_n\rangle \quad \text{where} \quad \begin{cases} x_1, \ldots, x_n \in X \\ r_1, \ldots, r_n \in [0,1] \text{ with } \sum_i r_i = 1. \end{cases}$$

The 'ket' notation $|x\rangle$ is meaningless syntactic sugar, used to distinguish elements $x \in X$ from their occurrence in such formal convex sums. Notice that such $\omega \in \mathcal{D}(X)$ can be identified with functions $\omega \colon X \to [0,1]$ with finite support $\mathrm{supp}(\omega) = \{x \in X \mid \omega(x) \neq 0\}$ and with $\sum_{x \in X} \omega(x) = 1$. This function-description is often more convenient.

We shall write $\mathcal{K}\ell(\mathcal{D})$ for the Kleisli category of the distribution monad $\mathcal{D}$. Its objects are sets X, and its morphisms $X \to Y$ are stochastic matrices, in the form of functions $X \to \mathcal{D}(Y)$.

We will see later (Section 3.1) how Bayesian networks can be seen as certain arrows of $\mathcal{K}\ell(\mathcal{D})$. For this interpretation, it is of importance that $\mathcal{K}\ell(\mathcal{D})$ forms a symmetric monoidal category, with the following ingredients. The monoidal product $\otimes$ is defined on objects as the cartesian product $\times$ in **Set**, with unit 1. On arrows $f \colon A \to X$ and $g \colon B \to Y$, it is defined as

$$f \otimes g \;:=\; \Big(A \times B \xrightarrow{f \times g} \mathcal{D}(X) \times \mathcal{D}(Y) \xrightarrow{\mathrm{dst}} \mathcal{D}(X \times Y) \Big)$$

where the map dst sends a pair $(\rho, \sigma) \in \mathcal{D}(X) \times \mathcal{D}(Y)$ to the distribution in $\mathcal{D}(X \times Y)$ given by $(x, y) \mapsto \rho(x) \cdot \sigma(y)$. The symmetry $\mathrm{tw}_{X,Y}$ on $X \times Y$ is the lifting to $\mathcal{K}\ell(\mathcal{D})$ of the isomorphism $X \times Y \xrightarrow{\cong} Y \times X$ in **Set**; we will omit the subscript when X and Y are clear from the context.

We now turn to the description of states and predicates in $\mathcal{K}\ell(\mathcal{D})$. Notice that states $\omega \colon 1 \to X$ in $\mathcal{K}\ell(\mathcal{D})$ can be identified with distributions $\omega \in \mathcal{D}(X)$. Since $\mathcal{D}(2) \cong [0,1]$ we can identify predicates $X \to 2 = 1 + 1$ in $\mathcal{K}\ell(\mathcal{D})$ with functions $X \to [0,1]$, that is, with fuzzy predicates. We will often make both identifications when emphasising the role of states and predicates in a computation.

Given a Kleisli map $f \colon X \to \mathcal{D}(Y)$, a state $\omega \in \mathcal{D}(X)$ and a predicate $q \in [0,1]^Y$ we have the following descriptions for state and predicate transformation. They arise from unravelling (Kleisli) composition in $\mathcal{K}\ell(\mathcal{D})$.

$$\begin{aligned} f_*(\omega) &:= \sum_{y \in Y} \big(\textstyle\sum_{x \in X} f(x)(y) \cdot \omega(x) \big) \big| y \big\rangle \\ f^*(q)(x) &:= \sum_{y \in Y} f(x)(y) \cdot q(y). \end{aligned} \tag{5}$$

For a distribution $\omega \in \mathcal{D}(X)$ and a predicate $p \in [0,1]^X$ on the same set X we define the validity $\omega \models p$ in $[0,1]$ as:

$$\omega \models p := \textstyle\sum_{x \in X} \omega(x) \cdot p(x). \tag{6}$$

If this validity $\omega \models p$ is non-zero, then the conditional state $\omega|_p \in \mathcal{D}(X)$ is given as

formal convex sum:

$$\omega|_p := \sum_{x \in X} \frac{\omega(x) \cdot p(x)}{\omega \models p} \big| x \big\rangle. \tag{7}$$

We shall describe a familiar medical test example in the current setting. We use the following notational convention. We write a letter D for a certain disease, which is represented as a two-element set $2_D = \{d, d^\perp\}$, where the element d represents occurrence of the disease, and $d^\perp$ represents non-occurrence. A distribution over 2_D is, *e.g.*, of the form $\frac{1}{4}\,|d\rangle + \frac{3}{4}\,|d^\perp\rangle$, when describing that the disease occurs with probability $\frac{1}{4}$. Similar we write 2_T for a (positive) test, where $2_T = \{t, t^\perp\}$. For each such set $2_A = \{a, a^\perp\}$ we write $A?\colon 2_A \to [0,1]$ for the sharp predicate given by $A?(a) = 1$ and $A?(a^\perp) = 0$.

Consider the following situation in the Kleisli category $\mathcal{K}\ell(\mathcal{D})$.

$$1 \xrightarrow{\ \omega\ } 2_D \xrightarrow{\ s\ } 2_T \qquad \text{with} \qquad \begin{cases} \omega = \frac{1}{100}\,|d\rangle + \frac{99}{100}\,|d^\perp\rangle \\ s(d) = \frac{9}{10}\,|t\rangle + \frac{1}{10}\,|t^\perp\rangle \\ s(d^\perp) = \frac{1}{20}\,|t\rangle + \frac{19}{20}\,|t^\perp\rangle\,. \end{cases}$$

The state ω describes the *a priori* probability of 1% that someone has the disease. The map s describes the sensitivity of the test: when someone has the disease, the test will be positive in 90% of the cases, and when someone does not have the disease there is still a 5% chance that the test is positive.

A basic question is: what is the chance that I have the disease if I test positive? We formalise this by adding the predicate $T?\colon 2_T \to [0,1]$, which expresses that there is a positive test. We then compute consecutively the predicate $s^*(T?)\colon 2_D \to [0,1]$, the validity $\omega \models s^*(T?)$ and the inferred conditional state $\omega|_{s^*(T?)}$. We use formulas (5), (6), and (7) for backward inference from Definition 2.1:

$$\begin{aligned} s^*(T?)(d) &= \tfrac{9}{10} \cdot 1 + \tfrac{1}{10} \cdot 0 \\ &= \tfrac{9}{10} \\ s^*(T?)(d^\perp) &= \tfrac{1}{20} \cdot 1 + \tfrac{19}{20} \cdot 0 \\ &= \tfrac{1}{20} \\ \omega \models s^*(T?) &= \tfrac{1}{100} \cdot \tfrac{9}{10} + \tfrac{99}{100} \cdot \tfrac{1}{20} \\ &= \tfrac{9}{1000} + \tfrac{99}{2000} \\ &= \tfrac{117}{2000} \\ \omega|_{s^*(T?)} &= \tfrac{2000}{117} \cdot \big(\tfrac{1}{100} \cdot \tfrac{9}{10}\,|d\rangle + \tfrac{99}{100} \cdot \tfrac{1}{20}\,|d^\perp\rangle \big) \\ &= \tfrac{18}{117}\,|d\rangle + \tfrac{99}{117}\,|d^\perp\rangle\,. \end{aligned} \tag{8}$$

Hence after a positive test the chance that I have the disease is $\frac{18}{117} \sim 15\%$. This is an instance of backward inference, where an observation on the codomain (the test outcome) changes the state of knowledge about the domain (the disease occurrence). Of course, standard Bayesian methods will arrive at the same outcome. The point is that we can describe these methods here in a uniform, abstract manner via calculations in (Kleisli) categories.

We briefly describe a forward example. Suppose that I know that the chance of having this disease is half as likely for me, for instance because I belong to a particular age group. We model this via the predicate $p\colon 2_D \to [0,1]$ given by $p(d) = \frac{1}{2}$ and $p(d^\perp) = 1$. We would like to learn what the probability is of getting a positive test under these circumstances.

We take a step back, and ask ourselves: what is the probability of getting a positive test in general — without the adapted likelihood. This probability is computed via the state transformer s_* from (5) — that is, via Kleisli composition in $\mathcal{K}\ell(\mathcal{D})$ as:

$$\begin{aligned} s_*(\omega) &= (\tfrac{1}{100} \cdot \tfrac{9}{10} + \tfrac{99}{100} \cdot \tfrac{1}{20}) \,|t\rangle + (\tfrac{1}{100} \cdot \tfrac{1}{10} + \tfrac{99}{100} \cdot \tfrac{19}{20}) \,|t^\perp\rangle \\ &= \tfrac{117}{2000} \,|t\rangle + \tfrac{1883}{2000} \,|t^\perp\rangle \,. \end{aligned}$$

For forward inference we first compute the conditional state $\omega|_p$ and then push it forward to a state $s_*(\omega|_p)$ on 2_T.

$$\begin{aligned} \omega \models p &= \tfrac{1}{100} \cdot \tfrac{1}{2} + \tfrac{99}{100} \cdot 1 \\ &= \tfrac{199}{200} \\ \omega|_p &= \tfrac{200}{199} \cdot \big(\tfrac{1}{100} \cdot \tfrac{1}{2} \,|d\rangle + \tfrac{99}{100} \cdot 1 \,|d^\perp\rangle\big) \\ &= \tfrac{1}{199} \,|d\rangle + \tfrac{198}{199} \,|d^\perp\rangle \\ s_*(\omega|_p) &= (\tfrac{1}{199} \cdot \tfrac{9}{10} + \tfrac{198}{199} \cdot \tfrac{1}{20}) \,|t\rangle + (\tfrac{1}{199} \cdot \tfrac{1}{10} + \tfrac{198}{199} \cdot \tfrac{19}{20}) \,|t^\perp\rangle \\ &= \tfrac{216}{3980} \,|t\rangle + \tfrac{3764}{3980} \,|t^\perp\rangle \,. \end{aligned}$$

Hence, upon knowing that I have a reduced (halved) risk, my chance of getting a positive test goes down from $\frac{117}{2000} \sim 5.8\%$ to $\frac{216}{3980} \sim 5.4\%$. The impact is limited, because I only have a very small chance of having the disease in the first place — and the false positive probability of the test is 5%.

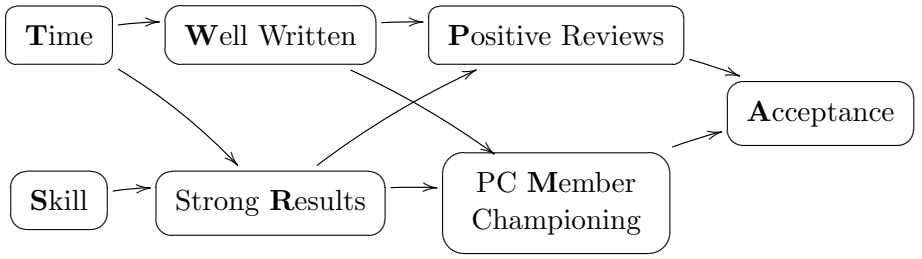
By imposing the predicate p on the disease state ω we adapt the influence of the state ω on the outcome. This may be useful for *counterfactual* reasoning, see [17]. In this way one can test to what extend a conclusion depends on certain initial states. For instance, if a particular conclusion is reached starting in a state where 70% of the participants is female, then by imposing an additional predicate on this state that changes the gender percentage, one can check if the same conclusion is reached.

3.1 Inference in a Bayesian network

Bayesian networks are graph-like structures, widely-adopted for the representation of probabilistic relationships between random events. They are usually depicted as directed acyclic graphs with nodes standing for random variables and edges indicating causal dependencies between them. Inference tasks are one of the fundamental uses of these networks. They are typically performed by updating a single node-event and then propagating the information to the rest of the network. Computing the inference typically goes through a repeated use of the Bayes' rule for conditional probability, see *e.g.* [16,18,17,2].

In this subsection we show how our abstract account of inference instantiates to the case of Bayesian networks. Our approach predicts the same outcomes as traditional Bayesian inference, but also improves it in two ways. First, it is more flexible and compositional, as it allows to focus on single nodes in the same way as on bigger portions of the network, with the same methodology. Second, it is more structured, in the sense that the computations that would require the use of Bayes' rule are carried out by the categorical machinery — essentially, by composition of arrows in a category.

In order to illustrate this picture, we will use as a running example the situation of a scientist that wants to publish a paper at a conference. The specification for the corresponding Bayesian network consists of a graph together with conditional probability tables.



(9)

$\Pr(T)$	$\Pr(S)$
$\frac{4}{10}$	$\frac{7}{10}$

T	$\Pr(W)$
t	$\frac{8}{10}$
f	$\frac{3}{10}$

T	S	$\Pr(R)$
t	t	$\frac{9}{10}$
t	f	$\frac{6}{10}$
f	t	$\frac{4}{10}$
f	f	$\frac{1}{10}$

W	R	$\Pr(P)$
t	t	$\frac{8}{10}$
t	f	$\frac{5}{10}$
f	t	$\frac{6}{10}$
f	f	$\frac{1}{10}$

W	R	$\Pr(M)$
t	t	$\frac{4}{10}$
t	f	$\frac{1}{10}$
f	t	$\frac{3}{10}$
f	f	0

P	M	$\Pr(A)$
t	t	1
t	f	$\frac{7}{10}$
f	t	$\frac{8}{10}$
f	f	$\frac{1}{10}$

The initial conditions of the example estimate whether there is enough time available to prepare the paper (the variable T) and whether the scientist is sufficiently skilled to do the necessary research (S). The results that the scientist is able to obtain (R) depend both on the time and the skill, while how well the paper reads only depends on the time. Both results and readability have an influence on whether the reviews will be positive (P), but results will be more relevant. Similarly, these two factors may lead a PC member to enthusiastically endorse the paper (M), independently of what the reviewers say, although this possibility is quite rare. Finally, acceptance (A) is influenced by the reviews and by the possible endorsement of a PC member.

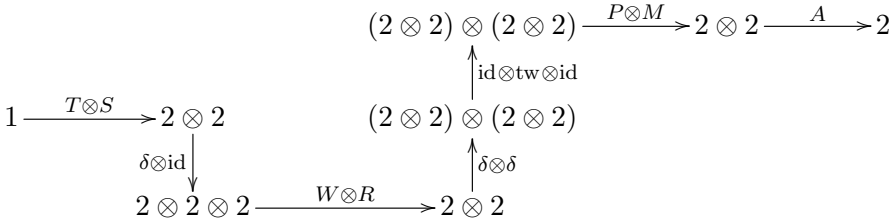
In order to study inference in this example, we first need to formulate it in more categorical terms. We shall express our Bayesian network (9) as an arrow in the Kleisli category $\mathcal{K}\ell(\mathcal{D})$ of the distribution monad $\mathcal{D}$. First, each node N of the graph, say with k incoming edges from nodes $N_1, N_2, \ldots, N_k$, is associated with an arrow $N\colon 2^k \to k$ in $\mathcal{K}\ell(\mathcal{D})$, which we conveniently write using the same labeling convention for the elements of 2 as in the disease example:

$$2_{N_1} \otimes 2_{N_2} \otimes \ldots \otimes 2_{N_k} \xrightarrow{\ N\ } 2_N.$$

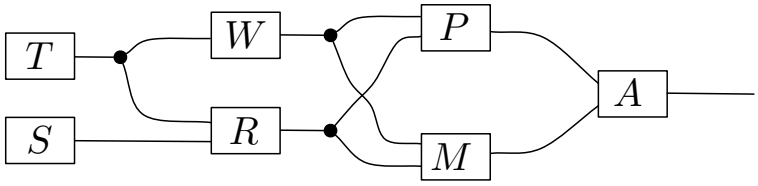
The probability distributions defining N are given according to the probability table of the node. For instance, the Kleisli map $A\colon 2_P \otimes 2_M \to 2_A$ for acceptance is defined by:

$$(p, m) \mapsto 1\,|a\rangle \qquad (p, m^\perp) \mapsto \tfrac{7}{10}\,|a\rangle + \tfrac{3}{10}\,|a^\perp\rangle$$
$$(p^\perp, m) \mapsto \tfrac{8}{10}\,|a\rangle + \tfrac{2}{10}\,|a^\perp\rangle \qquad (p^\perp, m^\perp) \mapsto \tfrac{1}{10}\,|a\rangle + \tfrac{9}{10}\,|a^\perp\rangle\,.$$

Another example is the initial map $T\colon 1 \to 2_T$ for the time node, which amounts to the distribution $\frac{4}{10}\,|t\rangle + \frac{6}{10}\,|t^\perp\rangle$ in $\mathcal{D}(2_T) \cong [0, 1]$. In order to recover the whole network (9), one pastes node-arrows together using the symmetric monoidal structure of $\mathcal{K}\ell(\mathcal{D})$, which we recalled in the beginning of this section. Nodes in (9) that have multiple outgoing edges are modeled by composing the corresponding arrow $2^k \to 2$ with the pairing map $\delta\colon 2 \to 2^2$ defined by $x \mapsto 1\,|(x, x)\rangle$. The Bayesian network (9) in its entirety is then expressed as the following arrow in $\mathcal{K}\ell(\mathcal{D})$, where for simplicity we omit the subscripts naming the elements of each copy of 2.



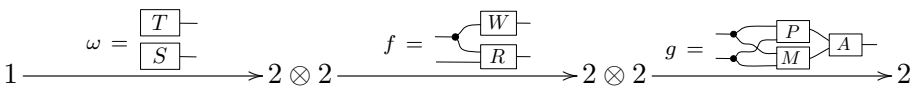
We have written the "structural" arrows vertically. A more insightful representation of the same arrow can be given using the graphical language of string diagrams [20], with 2^k depicted as a bundle of k wires and δ as  . The result almost resembles the original network.



(10)

It may be calculated [2] that the entire arrow $1 \to 2$ in (10) amounts to the distribution $0.48\,|a\rangle + 0.52\,|a^\perp\rangle$ in $\mathcal{D}(2) \cong [0, 1]$. In words: given 40% of chances that the scientist has enough time at disposal and 70% of chances of being adequately skilled, the odds of having a paper accepted at the conference is ∼48%.

We now have everything in place to instantiate our framework for inference. As this example is more elaborated than the previous ones, it gives us the possibility to explore the situation in which knowledge update only involves a segment of the computation, namely f or g in the following partitioned version of (10).



[2] For simplicity, here and in the next calculations we approximate distribution values to two decimal digits.

In order to formulate a backward inference question, we follow the recipe (3) and introduce a predicate $A?\colon 2_A \to [0,1]$ that tests for acceptance of the paper. It is a sharp predicate, defined by $A?(a) = 1$ and $A?(a^\perp) = 0$.

First we compute $\omega|_{(g\circ f)^*(A?)}$, that is, the odds that the accepted paper actually was submitted by a scientist with an adequate amount of time and skill to concoct it.

$$\begin{aligned}
\omega &= 0.28\,|t,s\rangle + 0.12\,|t,s^\perp\rangle + 0.42\,|t^\perp,s\rangle + 0.18\,|t^\perp,s^\perp\rangle \\
(g\circ f)^*(A?) &= \begin{cases} (t,s) \mapsto 0.67 & (t,s^\perp) \mapsto 0.58 \\ (t^\perp,s) \mapsto 0.40 & (t^\perp,s^\perp) \mapsto 0.29 \end{cases} \\
\omega \models (g\circ f)^*(A?) &= 0.48 \\
\omega|_{(g\circ f)^*(A?)} &= \sum_{x\in 2_T\otimes 2_S} \frac{\omega(x)\cdot (g\circ f)^*(A?)(x)}{0.48}\big|\,x\,\big\rangle \\
&= 0.39\,|t,s\rangle + 0.15\,|t,s^\perp\rangle + 0.35\,|t^\perp,s\rangle + 0.11\,|t^\perp,s^\perp\rangle
\end{aligned}$$

We observe that, after finding out that the paper has been accepted, the chances that the scientist had both sufficient time and skill rise from 28% to 39%.

As a second example, we shift the attention from the author to the paper itself. The following state on $2_W \otimes 2_R$ expresses the chances that an accepted paper was actually well written and contained strong scientific results. Note that it mixes state and predicate transformers to bind different segments of the network.

$$f_*(\omega)|_{g^*(A?)} = 0.48\,|w,r\rangle + 0.18\,|w,r^\perp\rangle + 0.24\,|w^\perp,r\rangle + 0.10\,|w^\perp,r^\perp\rangle$$

We see that, in our model, roughly one half of the accepted papers had both qualities, but only 10% of them had none.

Lastly, we consider an example of forward inference. Following the recipe (4), we introduce a predicate $E?\colon 2_T \otimes 2_S \to [0,1]$ on the state $\omega\colon 1 \to 2_T \otimes 2_S$: it expresses the event that, while writing the paper, the scientist finds out that the main result contains a minor mistake and thus needs some revision.

$$(t,s) \mapsto \tfrac{2}{10} \qquad (t,s^\perp) \mapsto \tfrac{4}{10} \qquad (t^\perp,s) \mapsto \tfrac{3}{10} \qquad (t^\perp,s^\perp) \mapsto \tfrac{6}{10}.$$

Differently from $A?$, this $E?$ is a fuzzy predicate: a mistake gets more likely the less time and skill are available to the scientist. If this situation occurs, the scientist may still be able to produce on time a paper that gets accepted, but chances are lower: they decrease from 48% to 43%. This is expressed by the following inference.

$$(g\circ f)_*(\omega|_{E?}) = 0.43\,|a\rangle + 0.57\,|a^\perp\rangle$$

Remark 3.1 We have modeled a Bayesian network as a graph in the Kleisli category $\mathcal{K}\ell(\mathcal{D})$. This is inspired by the approach of Fong [8], except that he uses the Kleisli category $\mathcal{K}\ell(\mathcal{G})$ of the Giry monad (even though all his examples are discrete). Such graphs in $\mathcal{K}\ell(\mathcal{D})$ or $\mathcal{K}\ell(\mathcal{G})$ can be seen as symmetric monoidal functors from a PROP $\mathcal{P}$, generated by a signature with the nodes and edges of the network, to the Kleisli category. We recall that a PROP (**pro**duct and **p**ermutation category [15]) is

a symmetric strict monoidal category with the natural numbers as objects and with monoidal product $\oplus$ given by addition of numbers. Intuitively, PROPs generalise Lawvere theories from the cartesian to the linear setting; functors from $\mathcal{P}$ as above are called the *models* of $\mathcal{P}$.

In our case, the model $\mathcal{P} \to \mathcal{K}\ell(\mathcal{D})$ sends $\oplus$ to the monoidal product $\otimes$ of $\mathcal{K}\ell(\mathcal{D})$, and sends the number 1 to the object $2 = 1 + 1$ in $\mathcal{K}\ell(\mathcal{D})$. $\mathcal{P}$ has pairing (copying) $\multimap\!\subset$, but a crucial point is that these copiers are not natural — as can be checked easily in $\mathcal{K}\ell(\mathcal{D})$. This implies that $\mathcal{P}$ is not a Lawvere theory (*cf.* [3]), and there is no associated monad on **Set**.

This monad perspective comes up in the following way. A Bayesian network with set of nodes X can be seen as a coalgebra of the form:

$$X \xrightarrow{c} \mathcal{B}(X) \qquad \text{where} \qquad \mathcal{B}(X) = \coprod_{U \subseteq_{\text{fin}} X} [0,1]^{2^{\#U}}$$

This coalgebra c sends a node $N \in X$ to a pair $c(N) = \langle c_1(N), c_2(N)\rangle$ where $c_1(N) \subseteq_{\text{fin}} X$ is a finite set of predecessor nodes of N, and $c_2(N)\colon 2^n \to [0,1]$ is the associated conditional probability table — where $n = \#c_1(N) \in \mathbb{N}$ is the number predecessors. Since $[0,1] \cong \mathcal{D}(2)$, this map $c_2(N)$ is a Kleisli map $2^n \to 2$ in $\mathcal{K}\ell(\mathcal{D})$, as used in the above description of the paper-acceptance example.

It is not hard to see that the mapping $X \mapsto \mathcal{B}(X)$ is a functor on **Set**, and comes with a unit map $X \to \mathcal{B}(X)$. But $\mathcal{B}$ is not a monad, at least not in the expected obvious sense, precisely because the copiers $\multimap\!\subset$ are not natural.

4 Inference with continuous probability

Our abstract description of inference allows us to transfer the definitions from the discrete to the continuous approach simply by switching from the Kleisli category $\mathcal{K}\ell(\mathcal{D})$ of the distribution monad to the Kleisli category $\mathcal{K}\ell(\mathcal{G})$ of the Giry monad [9] on measurable spaces. We shall sketch an example where the function f in the inference situation (3) is the identity, but where we have multiple predicates p_i for successive learning. Hence there is no predicate/state transformation involved. We describe the essentials and refer to [5] for more information.

A state $\omega\colon 1 \to X$ in the Kleisli category $\mathcal{K}\ell(\mathcal{G})$ is a probability measure $\omega \in \mathcal{G}(X)$, given by a function $\omega\colon \Sigma_X \to [0,1]$ that maps measurable subsets to probabilities. A predicate $p\colon X \to 2$ in $\mathcal{K}\ell(\mathcal{G})$ is a measurable function $p\colon X \to [0,1]$ since $\mathcal{G}(2) \cong [0,1]$. The validity $\omega \models p$ in $[0,1]$ and conditional state $\omega|_p$ in $\mathcal{G}(X)$ are given by the following integration formulas.

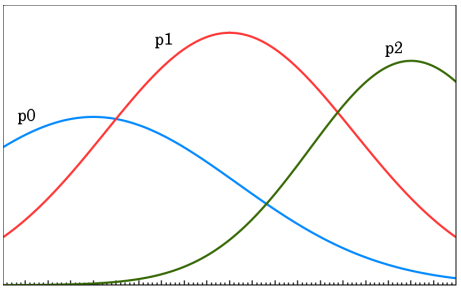
$$\omega \models p := \int p \,\mathrm{d}\omega \qquad \text{and} \qquad \omega|_p(M) := \frac{\int_M p \,\mathrm{d}\omega}{\omega \models p}. \tag{11}$$

Often the state/probability measure ω that we start from is given by a probability density function. This means that ω is of the form $\phi \models q$, for some predicate q. In that case the conditional state $\omega|_p = (\phi|_q)|_p$ is the same as the condition of the product predicate: $\phi|_{q \cdot p}$ with pdf $q \cdot p$. This greatly simplifies the picture below.

The inference example that we use is a continuous version of the archeological example described in [13]. The aim is to infer the date of a tomb at an archeological

site of which we already know that it is from the interval $0 - 100$ AD. We are specifically looking to find three kinds of objects, labelled $0, 1, 2$ of which we know the time of use more precisely. They are used to infer the age of the tomb. This knowledge is represented by three predicates $p_0, p_1, p_2 \colon [0, 100] \to [0, 1]$ given by the formulas:

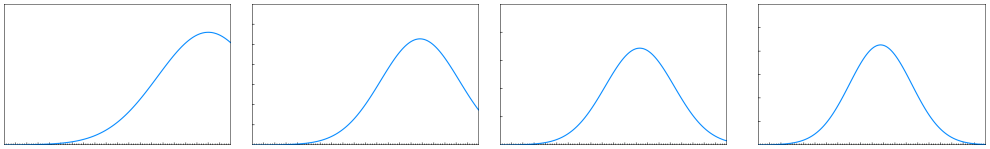
$$\begin{aligned} p_0(x) &= 0.6 \cdot e^{-(x-20)^2/2000} \\ p_1(x) &= 0.9 \cdot e^{-(x-50)^2/1500} \\ p_2(x) &= 0.8 \cdot e^{-(x-90)^2/1000} \end{aligned}$$



Our inference works as follows. We start from the uniform measure $\omega = \phi|_q$ with pdf $q(x) = \frac{1}{100}$ on $[0, 100]$, for the Lebesgue measure ϕ. Its probability on the sub-interval $[a, b] \subseteq [0, 100]$ is given by the integral:

$$\omega([a, b]) = \phi|_q([a, b]) = \int_a^b q \, \mathrm{d}\phi = \int_a^b \tfrac{1}{100} \, \mathrm{d}\phi = \tfrac{b-a}{100}.$$

We now successively observe objects $i_1, \ldots, i_n$, for $i_k = 0, 1, 2$, and compute the conditional probability measure $(\cdots(\omega|_{p_{i_1}})\cdots)|_{p_{i_n}}$. We can describe this measure via the product pdf $q \cdot p_{i_1} \cdots p_{i_n}$, after normalisation. Below we sketch the shape of some of the resulting pdf's (ignoring normalisation), after finding certain objects successively.



after finding 2 | after finding 2,1 | after finding 2,1,0 | after finding 2,1,0,0

These curves describe the inferred probability for the age of the tomb in the interval 0 – 100 AD.

5 Quantum inference

Our inference situations (3) and (4) can also be interpreted in the effectus of von Neumann algebras for quantum computation. Actually, one uses the opposite $\mathbf{vNA}^{\mathrm{op}}$ of the category $\mathbf{vNA}$ of von Neumann algebras, with normal completely positive unital maps between them (see [5] for details). We have to take the opposite category because maps between von Neumann algebras should be understood as predicate transformers. Typical examples are the von Neumann algebras $\mathcal{B}(\mathscr{H})$ of bounded operators on a Hilbert space $\mathscr{H}$. Below we use the matrix algebra $M_2 = \mathcal{B}(\mathbb{C}^2)$ as special case.

For instance, the situation (3) translates into a diagram of maps in the category

vNA pointing in the other direction:

$$\mathbb{C} \xleftarrow{\ \omega\ } \mathscr{B} \xleftarrow{\ f\ } \mathscr{A} \xleftarrow{\ q\ } \mathbb{C}^2$$

The conditional state $\omega|_{f^*(q)} \colon \mathscr{B} \to \mathbb{C}$ in backward inference is given by the general formula:

$$b \longmapsto \frac{\omega\big(\sqrt{f(q)} \cdot b \cdot \sqrt{f(q)}\big)}{\omega\big(f(q)\big)}. \tag{12}$$

In this situation predicate transformation $f^*(q) = f \circ q$ works in the opposite direction. The square-roots arise from the particular form of 'assert' map that is used for von Neumann algebras, see [5] for details. The predicate $q \colon \mathbb{C}^2 \to \mathscr{A}$ is a positive unital map, and can thus be identified with an effect in $\mathscr{A}$, that is, with an element $q \in \mathscr{A}$ satisfying $0 \leq q \leq 1$.

Bayesian inference in a quantum setting is a relatively new topic, see *e.g.* [14,6]. At this stage we only apply our general pattern from Definition 2.1 in a quantum setting. The illustration below repeats the disease-test example from Section 3 for the von Neumann algebra M_2 of 2×2 complex matrices. Our only ambition at this stage is to show how the quantum description extends the probabilistic one. Consider therefore the diagram:

$$\mathbb{C} \xleftarrow{\ \omega\ } M_2 \xleftarrow{\ s\ } M_2 \ni T? = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$$

These test (sensitivity) and state maps are given by:

$$s\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = \begin{pmatrix} \frac{9}{10}a + \frac{1}{10}d & 0 \\ 0 & \frac{1}{20}a + \frac{19}{20}d \end{pmatrix} \quad \text{and} \quad \omega\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = \tfrac{1}{100}a + \tfrac{99}{100}d.$$

Predicate transformation yields:

$$s(T?) = s\left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right) = \begin{pmatrix} \frac{9}{10} & 0 \\ 0 & \frac{1}{20} \end{pmatrix} \quad \text{and} \quad \omega\big(s(T?)\big) = \tfrac{117}{2000}.$$

The backward inferred state $\omega|_{s^*(T?)}$ is according to (12):

$$\begin{aligned} \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) &\longmapsto \tfrac{2000}{117} \cdot \omega\left(\begin{pmatrix} \sqrt{9/10} & 0 \\ 0 & \sqrt{1/20} \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \sqrt{9/10} & 0 \\ 0 & \sqrt{1/20} \end{pmatrix}\right) \\ &= \tfrac{2000}{117} \cdot \omega \begin{pmatrix} 9/10\, a & \sqrt{9/200}\, b \\ \sqrt{9/200}\, c & 1/20\, d \end{pmatrix} \\ &= \tfrac{18}{117}a + \tfrac{99}{117}d. \end{aligned}$$

We see that the outcome is the same, up to some re-shuffling, as in the discrete probabilistic presentation in (8). But this situation allows much richer structure, for instance using as state $\rho \colon M_2 \to \mathbb{C}$ the map $\rho\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = \frac{1}{2}(a - b - c + d)$.

Conclusions

This paper has clarified the role of states and predicates, and of state transformers and predicate transformers, in Bayesian inference. An abstract definition of for-

ward and backward inference has been given in the context of effectus theory, and interpreted and elaborated in several contexts and examples.

The generality of our approach allows for applications outside of the traditional probabilistic setting; the case of Von Neumann algebras is one such example which has been described here only in limited, probabilistic form. The power of the properly quantum approach (see also [14,6]) will be elaborated elsewhere.

The application to Bayesian networks also leaves room for interesting developments. As sketched in Remark 3.1, the interpretation of networks as arrows of $\mathcal{K}\ell(\mathcal{D})$ can be seen as part of a broader picture, that can be formulated in the language of PROPs and their models. We find particularly worthwhile trying to understand Bayesian inference, as introduced in the present paper, as a categorical transformation on models of a PROP: it should map one network into another one with the same topology, but different probability distributions.

Acknowledgements

The authors acknowledge support from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n^{o} 320571.

References

[1] Adams, R. and B. Jacobs, *A type theory for probabilistic and Bayesian reasoning* (2015), see arxiv.org/abs/1511.09230.

[2] Barber, D., "Bayesian Reasoning and Machine Learning," Cambridge Univ. Press, 2012.

[3] Bonchi, F., P. Sobocinski and F. Zanasi, *Lawvere categories as composed PROPs*, in: *Coalgebraic Methods in Computer Science (CMCS 2016), colocated with ETAPS 2016*, 2016, to appear.

[4] Borgström, J., A. Gordon, M. Greenberg, J. Margetson and J. V. Gael, *Measure transformer semantics for Bayesian machine learning*, Logical Methods in Comp. Sci. **9(3)** (2013), pp. 1–39.

[5] Cho, K., B. Jacobs, A. Westerbaan and B. Westerbaan, *An introduction to effectus theory* (2015), see arxiv.org/abs/1512.05813.

[6] Coecke, B. and R. Spekkens, *Picturing classical and quantum Bayesian inference.*, Synthese **186** (2012), pp. 651–696.

[7] Dijkstra, E. W., "A Discipline of Programming," Prentice Hall PTR, Upper Saddle River, NJ, USA, 1997, 1st edition.

[8] Fong, B., "Causal Theories: A Categorical Perspective on Bayesian Networks," Master's thesis, Univ. of Oxford (2012), see arxiv.org/abs/1301.6201.

[9] Giry, M., *A categorical approach to probability theory*, in: B. Banaschewski, editor, *Categorical Aspects of Topology and Analysis*, number 915 in Lect. Notes Math. (1982), pp. 68–85.

[10] Goodman, N. D., *The principles and practice of probabilistic programming*, in: *Proceedings of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '13 (2013), pp. 399–402.

[11] Jacobs, B., *New directions in categorical logic, for classical, probabilistic and quantum logic*, Logical Methods in Comp. Sci. **11(3)** (2015), pp. 1–76.

[12] Jacobs, B., *A recipe for state and effect triangles*, in: L. Moss and P. Sobocinski, editors, *Conference on Algebra and Coalgebra in Computer Science (CALCO 2015)*, LIPIcs **35** (2015), pp. 116–129.

[13] Jacobs, B., B. Westerbaan and A. Westerbaan, *States of convex sets*, in: A. Pitts, editor, *Foundations of Software Science and Computation Structures*, number 9034 in Lect. Notes Comp. Sci. (2015), pp. 87–101.

[14] Leifer, M. and R. Spekkens, *Towards a formulation of quantum theory as a causally neutral theory of Bayesian inference*, Phys. Rev. A **88(5)** (2013), p. 052130.

[15] Mac Lane, S., *Categorical algebra*, B Am Math Soc **71** (1965), pp. 40–106.

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

[17] Pearl, J., "Causality. Models, Reasoning, and Inference," Cambridge Univ. Press, 2009, 2nd ed. edition.

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

[19] Ścibior, A., Z. Ghahramani and A. Gordon, *Practical probabilistic programming with monads*, in: *Proc. 2015 ACM SIGPLAN Symp. on Haskell* (2015), pp. 165–176.

[20] Selinger, P., *A survey of graphical languages for monoidal categories*, Springer Lecture Notes in Physics **13** (2011), pp. 289–355, available at http://arxiv.org/abs/0908.3347.

[21] Staton, S., H. Yang, C. Heunen, O. Kammar and F. Wood, *Semantics for probabilistic programming: higher-order functions, continuous distributions, and soft constraints*, in: *Logic in Computer Science (LICS 2016)*, 2016, to appear, available at http://arxiv.org/abs/1601.04943.