Disintegration and Bayesian Inversion, Both Abstractly and Concretely

Kenta Cho[1] Bart Jacobs[2]

Institute for Computing and Information Sciences
Radboud University, Nijmegen, The Netherlands

Abstract

The notions of disintegration and Bayesian inversion are fundamental in conditional probability theory. They produce channels, as conditional probabilities, from a joint state, or from an already given channel (in opposite direction). These notions exist in the literature, in concrete situations, but are still insufficiently clear in general. This paper gives an abstract, pictorial description of both disintegration and Bayesian inversion, and relates them in general terms. Subsequently, more concrete descriptions are given in discrete and continuous probability. In the end, the paper illustrates the relevance of disintegration and inversion in probabilistic computation.

Keywords: Conditional probability, disintegration, Bayesian inversion, string diagrams

1 Introduction

The essence of conditional probability can be summarised informally in the following equation about probability distributions:

$$\textit{joint} = \textit{conditional} \cdot \textit{marginal}.$$

A bit more precisely, when we have joint probabilities $P(x, y)$ for elements x, y of a sample space:

$$P(y \mid x) \cdot P(x) = P(x, y) = P(x \mid y) \cdot P(y), \tag{1}$$

where $P(x)$ and $P(y)$ describe the marginals. We see that conditional probabilities can be constructed in two directions, namely y given x, and x given y.

Informally, disintegration gives a structural description of the above equation (1) in terms of states and channels, see Definition 2.1 below. In general terms, a *state* is a probability distribution of some sort (discrete, continuous, or even quantum) and a *channel* is a map or morphism in a probabilistic setting. It can take the form

[1] Email: K.Cho@cs.ru.nl
[2] Email: bart@cs.ru.nl

of a stochastic matrix, probabilistic transition system, Markov kernel, conditional probability table (in a Bayesian network), or morphism in a Kleisli category of a 'probabilistic' monad.

In this article we abstract away from these (interpretation) details and will describe disintegration pictorially, in the language of string diagrams. This language can be seen as the internal language of symmetric monoidal categories [18], with comonoid structure on objects, for discarding and copying. The essence of disintegration becomes: extracting a conditional probability channel from a joint state.

A categorical approach to Bayesian conditioning has appeared for instance in [5, 6, 20] and in [12–14]. The latter references use effectus theory [4, 11], a new comprehensive approach aimed at covering the logic of both quantum theory and probability theory, supported by a Python-based tool 'EfProb', for 'effectus probability'. It is used for the examples in Section 4.

Disintegration, also known as regular conditional probability, is a notoriously difficult operation in continuous probability (see *e.g.* [16]): it may not exist [21]; even if it exists it may be determined only up to negligible subsets; and it may not be continuous or computable [1]. Disintegration has been studied using categorical language in [5], where measure spaces are restricted to a certain class so that disintegration exists. It can also be handled via symbolic manipulation [19]. Our abstract pictorial formulation ignores these difficulties at first, in order to get a smooth general picture. The more subtle issues are addressed to some extent in Section 3. But really, our focus is mostly on the abstract formulation.

We thus describe disintegration as going from a joint state to a channel. A closely related concept is *Bayesian inversion*: it turns a channel (with a state) into a channel in opposite direction. We show how Bayesian inversion can be understood and expressed easily in terms of disintegration — and also how, in the other direction, disintegration can be obtained from Bayesian inversion.

Bayesian inversion is crucial for backward inference. We explain it informally: let σ be a state of a domain/type X, and $c\colon X \to Y$ be a channel; Bayesian inversion yields a channel $d\colon Y \to X$. Informally, it produces for an element $y \in Y$, seen as singleton/point predicate $\{y\}$, the conditioning of the state σ with the pulled back evidence $c^{-1}(\{y\})$.

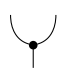
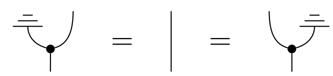
Bayesian inversion is relatively easy to define in discrete probability theory. The situation is much more difficult in continuous probability theory, first of all because point predicates $\{y\}$ do not make much sense there. A common solution is to assume a *likelihood*, that is, a probabilistic relation $X \times Y \to \mathbb{R}_{\geq 0}$. Such a likelihood gives rise to probability density function (pdf), providing a good handle on the situation, see [17]. The technical core of this paper is a generalisation of this likelihood-based approach to an effectus-theoretic setting, see Subsection 3.3.

The paper concludes in Section 4 at a very concrete level, by illustrating the usefulness of disintegration in probabilistic programming [3, 10, 15, 20]. Disintegration and Bayesian inversion are used to structurally organise state updates in presence of new evidence. In two examples the updated probabilities are calculated with EfProb, using its implementation of disintegration, based on the general description of Definition 2.1.

2 Disintegration, pictorially

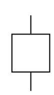
We start with a pictorial description of disintegration, because it is most intuitive. The pictures that we use are very elementary, using only relatively simple string diagrams. The flow is from bottom to top. States are written as triangles with multiple wires coming out on top, as on the left below.

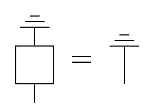
These wires correspond to connections, which can be put in parallel. They have types like X in the first one, but we often suppress such labels. Wires can be 'terminated', as written on the right above, using $\overline{\top}$. This corresponds to marginalisation of the state. Our calculus allows copying of wires, written as:

 and satisfying: 

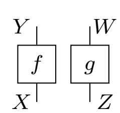
In a quantum setting discarding is possible, but not copying (because of 'no-cloning'). Both discarding and copying do exist in classical (non-quantum) probability, in the form of comonoid structure.

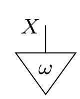
We need one further ingredient, namely a channel. It is a box with multiple incoming wires and multiple outgoing wires (zero, one, or more), as on the left below. In fact, a state is a special case of a channel, with zero incoming wires. We require channels (including states) to be *causal* in the sense that the equation on the right holds:





Interpretations of our graphical calculus will be described in Section 3, where we shall use Kleisli maps as semantics of channels. In technical terms, these string diagrams are a graphical language of suitable monoidal categories. We thus sometimes use the categorical (textual) notation as well, for example below, where $(\otimes, I)$ denotes the monoidal structure.

$X \otimes Z \xrightarrow{f \otimes g} Y \otimes W$ means 

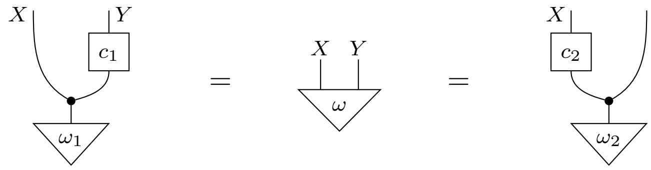
$I \xrightarrow{\omega} X$ means 

See [18] for further details.

We now introduce disintegration and Bayesian inversion, pictorially, in two separate definitions.

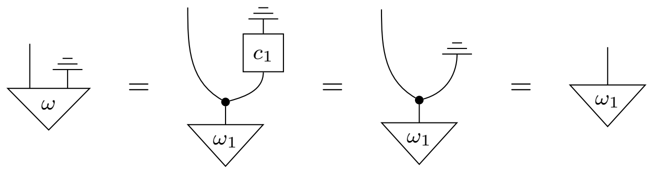
Definition 2.1 Disintegration involves extracting a channel from a joint state. We concentrate on the binary case, starting from a joint state ω, say with outgoing wires of types X, Y. *Disintegration in the first component* involves extracting a state of type X and a channel $c\colon X \to Y$ for which the equation on the left holds. *Disintegration in the second component* leads to a state of type Y and a channel

$d \colon Y \to X$ for which the second equation holds.



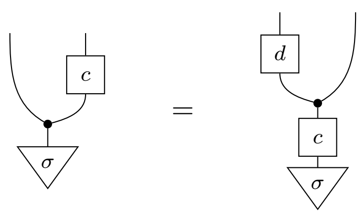
(2)

It is easy to see that the states ω_1 and ω_2 in (2) are necessarily marginals of ω, since



and similarly for ω_2.

Definition 2.2 Bayesian inversion is the process of going all the way from left to right (or the other way around) in (2). One then starts from a state σ of type X together with a channel $c \colon X \to Y$. They give rise to a joint state of type X, Y as on the left of (3) below. Disintegration in the second component then produces a channel $d \colon Y \to X$, which we call the *Bayesian inversion* for σ along c. It is easy to verify that d is required to satisfy:

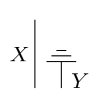


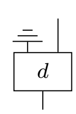
(3)

We see that disintegration involves 'sequentialising' a joint state, by extracting a channel. This channel need not be unique. It can also be viewed in terms of introducing 'causality' from X to Y, or from Y to X between the two types X, Y of the joint state. The essence of Bayesian inversion is inverting a channel $X \to Y$ to a channel $Y \to X$, in presence of a state of type X. As will be shown in the examples below, this is used for backward inference in a Bayesian setting, see also [14].

We introduced Bayesian inversion as a special case of disintegration. Conversely, disintegration (of a joint state) may be given by Bayesian inversion as follows.

Proposition 2.3 *Let ω be a joint state of type X, Y. If $d \colon X \to X \otimes Y$ is a Bayesian inversion for ω along the 'projection' channel $X \otimes Y \to X$ on the left below,*

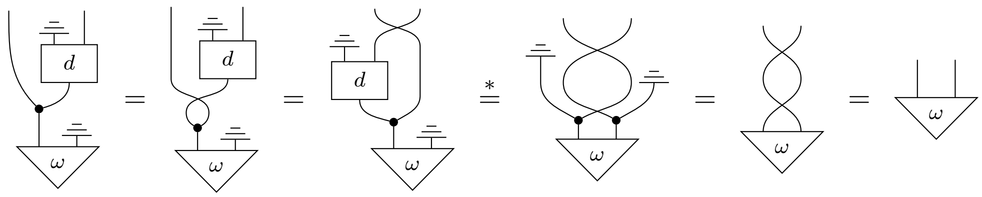




then d composed with discarder on X, shown on the right above, is a disintegration of ω in the first component.

The proof below uses crossing of wires: $\times$. In other words our graphical language is that of symmetric monoidal categories. We assume that each pair $(\curlyvee, \overline{\top})$ forms a commutative comonoid, and that the family of $(\curlyvee, \overline{\top})$ is suitably compatible (*e.g.* the copier on $X \otimes Y$ is formed by the copiers on X and Y). We again refer to [18] for more information.

Proof. Recall that the state of X given by disintegration must be the marginal of ω. Then:



The marked equality $\overset{*}{=}$ holds since d is a Bayesian inversion for ω along the projection $X \otimes Y \to X$. $\square$

In categorical language the disintegration equations (2) involve writing a joint state $\omega\colon I \to X \otimes Y$ as:

$$\begin{aligned} \omega &= (\mathrm{id} \otimes c_1) \circ \curlyvee \circ \omega_1 \qquad \text{where} \qquad \omega_1 = (\mathrm{id} \otimes \overline{\top}) \circ \omega \\ \omega &= (c_2 \otimes \mathrm{id}) \circ \curlyvee \circ \omega_2 \qquad \text{where} \qquad \omega_2 = (\overline{\top} \otimes \mathrm{id}) \circ \omega. \end{aligned} \tag{4}$$

Here we write $\curlyvee$ for the copier $A \to A \otimes A$ and $\overline{\top}\colon A \to I$ for the discarder.

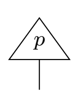
A Bayesian inversion d for a state σ along a channel c as in (3) satisfies:

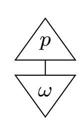
$$(\mathrm{id} \otimes c) \circ \curlyvee \circ \sigma = (d \otimes \mathrm{id}) \circ \curlyvee \circ c \circ \sigma. \tag{5}$$

2.1 *Disintegration, logically*

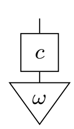
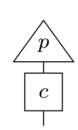
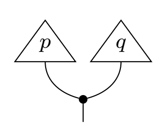
The pictorial approach to disintegration (and inversion) can also be extended with predicates. This is leads to formulations that are sometimes found in the literature. They will be briefly discussed here.

Predicates are 'up-side-down' states, without outgoing wire, as on the left below. They are not required to be causal; by definition, only discarders $\overline{\top}$ are causal predicates.



$\omega \models p \;:=$ 

On the right the *validity* $\omega \models p$ of predicate p in state ω is going from 'nothing' to 'nothing'. This is a scalar, which is generally written as $\langle s \rangle$. Next we need *state transformation* $c_*(\omega)$ and *predicate transformation* $c^*(p)$, and *conjunction* $p \,\&\, q$, which are defined as:

$c_*(\omega) \;:=$  $\qquad c^*(p) \;:=$  $\qquad p \,\&\, q \;:=$ 

The equation $c_*(\omega) \models p = \omega \models c^*(p)$ trivially holds, pictorially.

Proposition 2.4 *Let ω be a joint state of type X, Y, and let p be a predicate on X and q be a predicate on Y.*

(i) *Given disintegrations ω_i, c_i as in Definition 2.1, we have validity equations:*

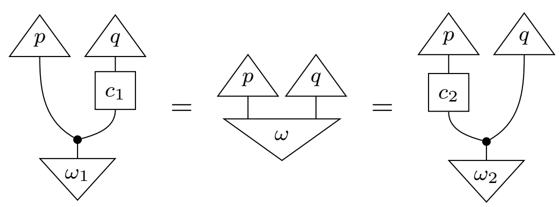
$$\omega_1 \models p \,\&\, c_1^*(q) \;=\; \omega \models p \otimes q \;=\; \omega_2 \models c_2^*(p) \,\&\, q.$$

(ii) *Starting from a state σ of X and a channel $c\colon X \to Y$, let $d\colon Y \to X$ be the associated Bayesian inversion, as in (3). Then we have a validity equation:*

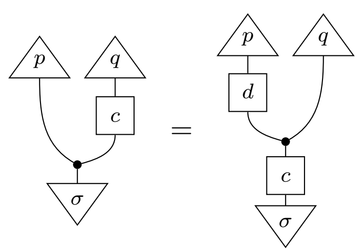
$$\sigma \models p \,\&\, c^*(q) \;=\; c_*(\sigma) \models d^*(p) \,\&\, q.$$

Sometimes, like in [6], the validity $\omega \models p \,\&\, q$ is written as a pairing $\langle p, q\rangle_\omega$. The equations in point (i) then look like an adjointness property: $\langle p, c_1^*(q)\rangle_{\omega_1} = \langle c_2^*(p), q\rangle_{\omega_2}$.

Proof. These are immediate consequences from the definitions of disintegration and inversion. The equations in point (i) amount to:



whereas the equation in point (ii) is:



□

3 Semantics of disintegration pictures

So far we have looked at pictures of disintegration and inversion. An urgent question is: can we give meaning to these pictures (states, channels, but also predicates) in some mathematical universe. We shall do so in two Kleisli categories $\mathcal{K}\ell(\mathcal{D})$ and $\mathcal{K}\ell(\mathcal{G})$, namely of the distribution monad $\mathcal{D}$ for discrete probability, and of the Giry monad $\mathcal{G}$ for continuous probability. The former is easier, and the latter corresponds to the ordinary notion of disintegration and inversion in the literature. After that, we turn to a general axiomatised setting. We refer to [12] for more information about these monads and for notation.

3.1 Disintegration in $\mathcal{K}\ell(\mathcal{D})$

A *channel* in probabilistic computation is an arrow in the Kleisli category $\mathcal{K}\ell(\mathcal{D})$ of the distribution monad $\mathcal{D}$. A channel $X_1, \ldots, X_n \to Y_1, \ldots, Y_m$ between sets X_i, Y_j is a Kleisli map $X_1 \times \cdots \times X_n \to \mathcal{D}(Y_1 \times \cdots \times Y_m)$. In particular, a *state* of type X is a Kleisli map $1 \to \mathcal{D}(X)$, which can simply be identified with a distribution $\omega \in \mathcal{D}(X)$. Such a distribution is given by a probability mass function $\omega \colon X \to [0,1]$ with finite support and $\sum_x \omega(x) = 1$. It is sometimes written as formal convex sum $\sum_x \omega(x)|x\rangle$.

Let $\omega \in \mathcal{D}(X \times Y)$ be a joint state, as in the middle of the disintegration diagram (2). The two marginals $\omega_1 \in \mathcal{D}(X)$ and $\omega_2 \in \mathcal{D}(Y)$ are given by:

$$\omega_1(x) = \textstyle\sum_y \omega(x,y) \qquad \text{and} \qquad \omega_2(y) = \textstyle\sum_x \omega(x,y).$$

We unravel the two equations in (2). Two channels $c_1 \colon X \to \mathcal{D}(Y)$ and $c_2 \colon Y \to \mathcal{D}(X)$ are disintegrations of ω, in the first and second component respectively, if the following two equations hold, for all $x \in X, y \in Y$.

$$c_1(x)(y) \cdot \omega_1(x) = \omega(x,y) = c_2(y)(x) \cdot \omega_2(y).$$

This means that if $\omega_1(x)$ and $\omega_2(y)$ are non-zero, the disintegration channels c_1, c_2 are completely determined as:

$$c_1(x)(y) = \frac{\omega(x,y)}{\omega_1(x)} \qquad \text{and} \qquad c_2(y)(x) = \frac{\omega(x,y)}{\omega_2(y)}$$

This shows that disintegration is nothing but the conditional probability calculated by the ordinary formula $P(y \mid x) \cdot P(x) = P(x,y) = P(x \mid y) \cdot P(y)$.

If $\omega_1(x) = 0$ or $\omega_2(y) = 0$, then $\omega(x,y) = 0$, so that $c_1(x)$ and $c_2(y)$ can be chosen arbitrarily. We conclude that disintegrations always exist in discrete probability, as formalised in $\mathcal{K}\ell(\mathcal{D})$.

We briefly describe Bayesian inversion in $\mathcal{K}\ell(\mathcal{D})$. Let state $\sigma \in \mathcal{D}(X)$ and channel $c \colon X \to \mathcal{D}(Y)$ be given, as in (3). The inversion $d \colon Y \to \mathcal{D}(X)$ is then described by *Bayes' formula*:

$$d(y)(x) = \sigma|_{c^*(\mathbf{1}_{\{y\}})}(x) = \frac{c(x)(y) \cdot \sigma(x)}{\sum_{x'} c(x')(y) \cdot \sigma(x')}. \tag{6}$$

This assumes that the denominator is not zero. It is easy to verify that this definition of the channel d then makes equation (3) true. The term in the middle of Eq. (6) emphasises that inversion can be understood as backwards learning, in the sense of [14], where a state is updated with a transformed predicate.

3.2 Disintegration in $\mathcal{K}\ell(\mathcal{G})$

We move on to a continuous (measure-theoretic) setting using the Kleisli category $\mathcal{K}\ell(\mathcal{G})$ of the Giry monad $\mathcal{G}$. Channels are measurable functions of the form $X \to \mathcal{G}(Y)$, where X and Y are measurable spaces. They are often called probability kernels. States are then identified with probability measures $\sigma \in \mathcal{G}(X)$.

Let $\omega \in \mathcal{G}(X \times Y)$ be a joint state on the product $X \times Y$ of measurable spaces. The marginal $\omega_1 \in \mathcal{G}(X)$ is then given by $\omega_1(A) = \omega(A \times Y)$, for $A \in \Sigma_X$. It is straightforward to check that a channel $c_1 \colon X \to \mathcal{G}(Y)$ is a disintegration of ω, *i.e.* satisfies Eq. (2) if and only if for all $A \in \Sigma_X, B \in \Sigma_Y$

$$\omega(A \times B) = \int_A c_1(x)(B)\, \omega_1(\mathrm{d}x)$$

This is the ordinary notion of disintegration, or regular conditional probability. [3] For example it can be found in [7] and [1].

To turn to Bayesian inversion in $\mathcal{K}\ell(\mathcal{G})$, let $\sigma \in \mathcal{G}(X)$ be a state and $c \colon X \to \mathcal{G}(Y)$ a channel. Then a channel $d \colon Y \to \mathcal{G}(X)$ is a Bayesian inversion, *i.e.* satisfies Eq. (3), if and only if

$$\int_A c(x)(B)\, \sigma(\mathrm{d}x) = \int_B d(y)(A)\, c_*(\sigma)(\mathrm{d}y) \tag{7}$$

In the literature it is often formulated for a 'deterministic' channel c only, namely for a measurable function $f \colon X \to Y$ rather than a kernel. In that case Eq. (7) becomes:

$$\sigma(A \cap f^{-1}(B)) = \int_B d(y)(A)\, \sigma(f^{-1}(\mathrm{d}y))$$

This formulation is found in [2], and is equivalent to "regular conditional probability consistent with f" defined in [9, 452E].

The equations above may be understood logically in the fashion of Subsection 2.1, via the identification of the measurable subsets A, B with the corresponding indicator functions/predicates $\mathbf{1}_A, \mathbf{1}_B$. For example, Eq. (7) can be rewritten as below, being a special case of Proposition 2.4 (ii).

$$\sigma \models \mathbf{1}_A \,\&\, c^*(\mathbf{1}_B) \;=\; c_*(\sigma) \models d^*(\mathbf{1}_A) \,\&\, \mathbf{1}_B$$

Note that predicates live in $\mathcal{K}\ell(\mathcal{G}')$ rather than $\mathcal{K}\ell(\mathcal{G})$, where $\mathcal{G}' = \mathcal{G}((-)+1)$ is the subprobability Giry monad (see [12]).

Disintegration or Bayesian inversion is much harder in continuous probability than in discrete probability. In fact, disintegration may not even exist [21], though it does exist for standard Borel spaces. A detailed analysis of the existence is found in [7]. We thus need to restrict ourselves to a certain subcategory of $\mathcal{K}\ell(\mathcal{G})$ if we want disintegration to always exist (*cf.* [5]).

Rather than dealing with the general situation, we here concentrate on Bayesian inversion in a special but common situation where a channel is given via probability densities or likelihoods. This situation will be considered in abstract form in Subsection 3.3 below. For now we assume that the channel $c \colon X \to \mathcal{G}(Y)$ is given by an integral as $c(x)(B) = \int_B \ell(x,y)\, \tau(\mathrm{d}y)$ for some measurable function $\ell \colon X \times Y \to \mathbb{R}_{\geq 0}$ and some measure τ on Y. Typically Y is the real line and τ is the Lebesgue measure. The value $\ell(x,y)$ is a probability density of y given x, and often called a *likelihood* of x given y. In this situation we can calculate a Bayesian

[3] We note that in the literature there are subtly different definitions of disintegration and regular conditional probability, depending on the precise assumptions that are being used.

inversion $d\colon Y \to \mathcal{G}(X)$ by:

$$d(y)(A) = \frac{\int_A \ell(x,y)\,\sigma(\mathrm{d}x)}{\int_X \ell(x,y)\,\sigma(\mathrm{d}x)} \tag{8}$$

Observe the analogy with Bayes' formula (6) for discrete probability. When the denominator is zero, $d(y)$ may be chosen arbitrarily. It is not hard to check that d defined in this way satisfies (7).

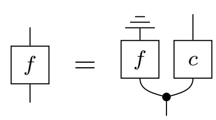
Having densities or likelihoods is a realistic assumption in probabilistic programming [3, 20]. It is also assumed in EfProb used in Section 4, so we can calculate disintegration and Bayesian inversion there. Moreover, the calculation of Bayesian inversion based on likelihoods (8) can be generalised to a categorical abstract setting. This is explained in the next subsection.

3.3 Inversion for 'probability' monads

Finally we shall investigate disintegration in a general axiomatised setting of 'probability' monads, following [12], along the lines of effectus theory [4]. We here describe the setting briefly and refer to [12] for further details.

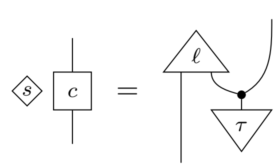
Let T be an affine commutative monad on a distributive category $\mathbf{C}$, and $T' = T((-)+1)$ the 'lifted' monad on $\mathbf{C}$. Then T' is a commutative monad too and hence the Kleisli category $\mathcal{K}\ell(T')$ is a symmetric monoidal category. There is moreover a commutative comonoid structure $X \to X \otimes X$ and $X \to I$ for each object X, which comes from the cartesian structure of $\mathbf{C}$. The category $\mathcal{K}\ell(T')$ is therefore suitable to interpret string diagrams in Section 2, and we have notions of states, channels, predicates, disintegration and inversion there.

We make one more assumption, *normalisation*, described below. Note that $\mathcal{K}\ell(T')$ has zero arrows $0_{XY}\colon X \to Y$ given by the obvious composite $X \to 1 \to Y + 1 \to T(Y+1)$ in $\mathbf{C}$. We say that an arrow $f\colon X \to Y$ is *nowhere zero* if $f_*(\omega) = f \circ \omega$ is nonzero for all states ω of X. Then our assumption is: for any nowhere zero arrow $f\colon X \to Y$, there is a unique channel $c\colon X \to Y$ (*i.e.* an arrow with $\overline{\top} \circ c = \overline{\top}$) such that



We call c the normalisation of f and denote it by $\mathrm{nrm}(f)$. In particular we have *normalisation for states*, which says: for any nonzero arrow $f\colon I \to X$ there is a unique state $\omega\colon I \to X$ with $f = (\overline{\top} \circ f) \cdot \omega$. It follows that: if $s \cdot \omega = s \cdot \omega'$ for states ω, ω' and a nonzero scalar s, then $\omega = \omega'$.

Definition 3.1 We say a channel $c\colon X \to Y$ is represented by a predicate ℓ on $X \otimes Y$ with respect to a state τ of Y if

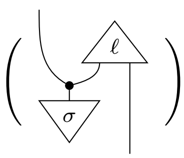
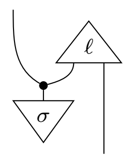


(9)

for some nonzero scalar s. We call ℓ a *likelihood relation* for channel c with respect to state τ.

The definition is motivated by the calculation of a Bayesian inversion in Subsection 3.2, in a situation where a channel is given via a likelihood function $\ell\colon X \times Y \to \mathbb{R}_{\geq 0}$. The equation (9) corresponds to the normalisation condition $\int_Y \ell(x,y)\,\tau(\mathrm{d}y) = 1$, but here we require this only up to a scalar s.

Now we consider a state σ of X, and a channel $c\colon X \to Y$ represented by a likelihood relation ℓ with respect to τ as in (9) above. We can form a channel $d\colon Y \to X$ in the other direction, via normalisation:

$d \;\coloneqq\; \mathrm{nrm}$  assuming that  is nowhere zero.

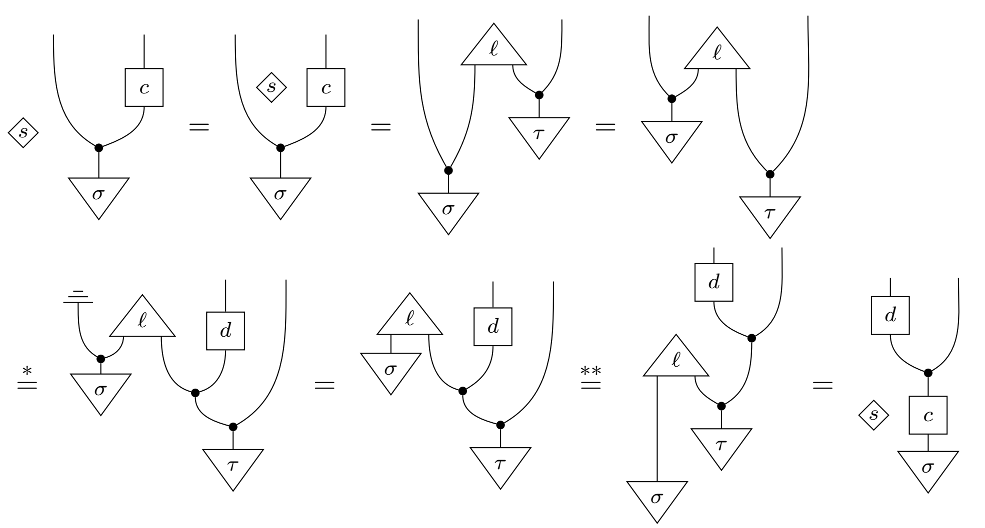
We note that the nowhere zeroness here amounts to assuming that the denominator of (8) is nonzero for all y.

We can now formulate and prove our main technical result.

Theorem 3.2 *In the setting described above, $d\colon X \to Y$ is a Bayesian inversion for σ along $c\colon X \to Y$. Namely, they satisfy Eq. (3).*

The proof below uses standard rules for string diagrams, as internal language for symmetric monoidal categories, see [18].

Proof. Let s be a nonzero scalar, as in Eq. (9). Then:



where $\overset{*}{=}$ follows by the definition of d, and $\overset{**}{=}$ by the associativity of copying $\curlyvee$. We then obtain Eq. (3) by uniqueness of normalisation. $\square$

4 Disintegration in probabilistic computation

This section illustrates the usefulness of disintegration and inversion in probabilistic computation, via two examples: one discrete, and one mixture of discrete and continuous. The formulation and actual computation will be done via the new EfProb tool[4] for probabilistic calculations. It includes support for disintegration, which is calculated from probability densities / likelihoods in the continuous case.

Diseases and moods

We consider a person whose state is described in terms of having a disease (or not), and have a good mood (or not). This is captured as a joint state, in which the disease and the mood are correlated: a high likelihood of disease corresponds to a low mood. There is a medical test available for the disease. We ask ourselves the question: how does a positive test influence the mood?

We formalise this example in the EfProb language. The joint (prior) state is defined as follows, starting with the relevant domains (types).

```
>>> disease_dom = ['D', '~D']
>>> mood_dom = ['M', '~M']
>>> w = State([0.05, 0.5, 0.4, 0.05], [disease_dom, mood_dom])
0.05|D,M> + 0.5|D,~M> + 0.4|~D,M> + 0.05|~D,~M>
```

The last line is most relevant. It describes the probabilities of the prior joint state `w` associated with the four combinations of disease `D` or not `~D`, and (good) mood `M` or not `~M`.

We can concentrate on the disease or mood separately, via marginalisations. In EfProb this is done as follows, via post-fix selection operations.

```
>>> w1 = w % [1,0]
>>> w2 = w % [0,1]
>>> w1
0.55|D> + 0.45|~D>
>>> w2
0.45|M> + 0.55|~M>
```

The sensitivity of the test is defined as a channel. If the disease is present, the test gives a positive outcome in 90% of the cases. But if the disease is absent, the test still has a 5% change of being positive. This is captured in the definition of the sensitivity channel below. It is applied to the (marginal) of the prior state to see what the a priori likelihood of a positive test is. This is done via state transformation, which is written in EfProb as >>.

```
>>> sensitivity = chan_from_states([flip(9/10), flip(1/20)], disease_dom)
>>> sensitivity >> w1
0.518|True> + 0.482|False>
```

[4] EfProb is developed by the authors and is publicly available from `efprob.cs.ru.nl`.

As explained above, we are interested in the mood after a positive test. This requires updating the prior state. Below we first define the positive-test predicate, and then condition (update, revise) the state (see [12] for details about conditioning), via the EfProb-notation `/`. We introduce a positive-test predicate `pos_test` via predicate transformation, written as << in EfProb. In order to use it for conditioning the joint state `w`, we have to 'weaken' (extend) the predicate `pos_test` to the whole domain of `w`. This is done via parallel conjunction `@` with the truth predicate. Finally, the second marginal of the updated state `s` gives the new mood, after the test.

```
>>> pos_test = sensitivity << yes_pred
>>> s = w / (pos_test @ truth(mood_dom))
>>> s % [0,1]
0.126|M> + 0.874|~M>
```

Clearly, a positive test leads to a lower mood.

Next we show how this result can also be obtained via disintegration of the prior joint state `w`. There are two ways to do this, via disintegration in the first component and state transformation, or via disintegration in the second component and predicate transformation. We describe them both.

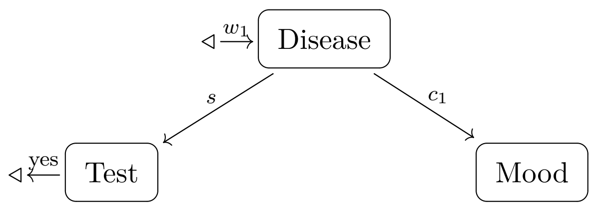
```
>>> c1 = w.disintegration([1,0])
>>> c1 >> (w1 / pos_test)
0.126|M> + 0.874|~M>
```

Via disintegration in the second component we get the same result:

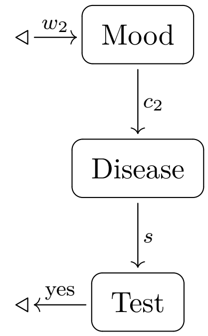
```
>>> c2 = w.disintegration([0,1])
>>> w2 / (c2 << pos_test)
0.126|M> + 0.874|~M>
```

The fact that these three approaches lead to the same mood distribution follows from a more general result. It can be proven using the disintegration equations (4), and some general facts about conditioning. This goes beyond the scope of the present paper.

We conclude this example by pointing out that disintegration is the key technique for turning a (possibly large) joint state into a Bayesian network. This is done below for the joint disease-mood state — where we abbreviate the sensitivity channel as s. Notice that the disintegrations in the two different components give channels c_1 and c_2 in different directions, and thus give different graphs. The states and predicates (priors and evidence) are added as side-information, via the small triangles. Such Bayesian networks can be interpreted as graphs in the Kleisli category of the distribution monad, see *e.g.* [14] and [8]. The two computations of the posterior mood state via disintegration, as described above in EfProb, are repeated below the networks in mathematical notation, where $(-)_*$ and $\gg$ are used for state transformation, and $(-)^*$ and $\ll$ for predicate transformation, and $|$ describes state update (conditioning).



$(c_1)_*\big(w_1|_{s^*(\text{yes})}\big)$



$w_2|_{(c_2)^*(s^*(\text{yes}))}$

Customers calling

Imagine a call centre that is open for 8 hours on each day of the week. The distribution of calls is different on weekends (Sat-Sun) from other days (Mon-Fri). What can we then learn from a single call at a given time of the day regarding whether it is weekend or not?

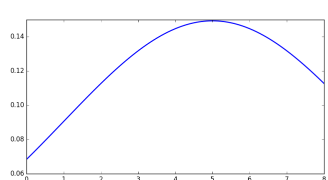
The formalisation in EfProb starts by defining a domain with label `W` for weekend and `~W` for non-weekend.

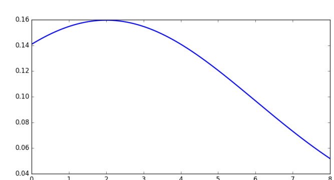
```
>>> weekend_dom = ['W', '~W']
>>> prior = State([2/7, 5/7], weekend_dom)
>>> prior
0.286|W> + 0.714|~W>
```

Next we have a channel that assigns a different Gaussian distribution to `W` and to `~W`.

```
>>> hours_dom = R(0,8)
>>> c = chan_from_states([gaussian_state(5,4,hours_dom),
...                        gaussian_state(2,4,hours_dom)], weekend_dom)
```

The probability density functions of the two distributions look as follows.





In the weekend diagram on the left we see that the calls start coming in later. Now we ask ourselves the question: suppose we see one call at (hour) 6. How does this affect the prior distribution? Of course, the updated distribution should have a higher likelihood for 'weekend' since 6 is relatively late.

We construct an inversion `d` of the above channel `c` in order to compute the updated distribution. We first form a joint state `w` as on the left in (3), and then we disintegrate it in its second component. It yields the channel `d`, going from `hours_dom` to `weekend_dom`. The distribution at time 6 is obtained by applying channel `d` to the value 6.

```
>>> w = (idn(weekend_dom) @ c) >> (copy(weekend_dom) >> prior)
>>> d = w.disintegration([0,1])
>>> d(6)
0.374|W> + 0.626|~W>
```

We see that the weekend probability has increased indeed.

5 Conclusions

The main contributions of this paper are:

- an abstract but intuitive formulation of disintegration and Bayesian inversion in the pictorial language of string diagrams;
- an interpretation of these constructs in the Kleisli categories of the distribution and Giry monad, culminating in an effectus-theoretic interpretation with a construction of inversion via likelihoods, see Theorem 3.2;
- an illustration of the use of disintegration and inversion for channel-based probability calculations, via the implementation of disintegration in the tool EfProb.

References

[1] Ackerman, N. L., C. E. Freer and D. M. Roy, *Noncomputable conditional distributions*, in: *Logic in Computer Science* (2011), pp. 107–116.

[2] Ackerman, N. L., C. E. Freer and D. M. Roy, *On computability and disintegration*, Mathematical Structures in Computer Science (2016), pp. 1–28.

[3] 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.

[4] Cho, K., B. Jacobs, A. Westerbaan and B. Westerbaan, *An introduction to effectus theory* (2015), preprint. arXiv:1512.05813 [cs.LO].

[5] Culbertson, J. and K. Sturtz, *A categorical foundation for Bayesian probability*, Applied Categorical Structures **22** (2014), pp. 647–662.

[6] Dahlqvist, F., V. Danos, I. Garnier and O. Kammar, *Bayesian inversion by ω-complete cone duality*, in: *CONCUR*, LIPIcs **59** (2016), pp. 1:1–1:15.

[7] Faden, A. M., *The existence of regular conditional probabilities: Necessary and sufficient conditions*, The Annals of Probability **13** (1985), pp. 288–298.

[8] Fong, B., "Causal Theories: A Categorical Perspective on Bayesian Networks," Master's thesis, Univ. of Oxford (2012), arXiv:1301.6201 [math.PR].

[9] Fremlin, D. H., "Measure Theory (5 volumes)," Torres Fremlin, 2000.

[10] Gordon, A., T. Henzinger, A. Nori and S. Rajamani, *Probabilistic programming*, in: *Future of Software Engineering* (2014), pp. 167–181.

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

[12] Jacobs, B., *From probability monads to commutative effectuses*, Journ. of Logical and Algebraic Methods in Programming **156** (2017, to appear).

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

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

[15] Kaminski, B., J.-P. Katoen, C. Matheja and F. Olmedo, *Weakest precondition reasoning for expected run-times of probabilistic programs*, in: *European Symp. on Programming*, Lect. Notes Comp. Sci. **9632** (2016), pp. 364–389.

[16] Panangaden, P., "Labelled Markov Processes," Imperial College Press, 2009.

[17] Pawitan, Y., "In All Likelihood. Statistical Modelling and Inference Using Likelihood," Clarendon Press, 2001.

[18] Selinger, P., *A survey of graphical languages for monoidal categories*, in: *New Structures for Physics*, Lecture Notes in Physics **813**, Springer, 2010 pp. 289–355.

[19] Shan, C.-c. and N. Ramsey, *Exact Bayesian inference by symbolic disintegration*, in: *Princ. of Programming Languages* (2017), pp. 130–144.

[20] 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* (2016), pp. 525–534.

[21] Stoyanov, J. M., "Counterexamples in Probability," Dover, 2014, third edition.