The Compositional Structure of Bayesian Inference

Dylan Braithwaite
Department of Computer and Information Sciences, University of Strathclyde, Glasgow, UK

Jules Hedges
Department of Computer and Information Sciences, University of Strathclyde, Glasgow, UK

Toby St Clere Smithe
Topos Institute, Berkeley, CA, USA
Department of Experimental Psychology, University of Oxford, UK

Abstract

Bayes' rule tells us how to invert a causal process in order to update our beliefs in light of new evidence. If the process is believed to have a complex compositional structure, we may observe that the inversion of the whole can be computed piecewise in terms of the component processes. We study the structure of this compositional rule, noting that it relates to the lens pattern in functional programming. Working in a suitably general axiomatic presentation of a category of Markov kernels, we see how we can think of Bayesian inversion as a particular instance of a state-dependent morphism in a fibred category. We discuss the compositional nature of this, formulated as a functor on the underlying category and explore how this can used for a more type-driven approach to statistical inference.

2012 ACM Subject Classification Theory of computation → Categorical semantics; Mathematics of computing → Probabilistic representations; Mathematics of computing → Bayesian computation

Keywords and phrases monoidal categories, probabilistic programming, Bayesian inference

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.24

Related Version This paper contains material from unpublished preprints [16] and [3].
Extended Version: https://arxiv.org/abs/2305.06112

Funding *Dylan Braithwaite*: This author is funded through an NPL Industrial CASE Studentship.

1 Introduction

A *Markov kernel* f is a function whose output is stochastic given the input. It can be thought of as a stateless device which, if supplied with inputs, will produce an output that depends probabilistically on the input. Mathematically speaking, this is nothing but a conditional distribution $\mathbb{P}(y|x)$. The problem of Bayesian inversion is that we observe the output of such a device but only have a probabilistic "prior" belief about the input, and we would like to update our beliefs about what the input was given the output. For example, suppose a process takes the roll of an unseen die and tells us whether it was even or odd, except with probability 10% it flips the result. If the process tells us that a specific roll resulted in "even" then Bayes' law tells us how we should update a prior belief that the die is uniformly random, to obtain a belief about what the specific roll was.

It is commonly the case that the process is not a black box, but is a composite process formed from simpler pieces. The conditional distribution formed by sequentially composing two such stochastic functions is given by forming their joint distribution and then marginalising over the middle variable. In the case of a long chain of sequentially composed functions, belief updating for the whole can be done "compositionally" in terms of belief updating for each individual part, in a process notably similar to backpropagation in neural networks. A process such as this underlies probabilistic programming languages, which are able to run programs "backwards" after conditioning on some output.

48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023).
Editors: Jérôme Leroux, Sylvain Lombardy, and David Peleg; Article No. 24; pp. 24:1–24:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The goal of this paper is to study this process of compositional Bayesian inversion of Markov kernels in isolation, using a suitable axiomatisation of a category of Markov kernels. We provide a method to build categories whose morphisms are pairs of a Markov kernel and an associated "Bayesian inverter", which is itself built compositionally.

Symmetric monoidal categories with compatible families of copy and delete morphisms have been identified as an expressive language for synthetically representing concepts from probability theory [5, 8]. The typical interpretation given is that the objects represent sets or measurable spaces, and morphisms are Markov kernels between them. Branded *Markov categories* [8] in this context, these categories have recently seen widespread use in applied category theory as a foundation of probability for two reasons: they allow working axiomatically while abstracting over the specific details of theories such as stochastic matrices, Gaussian kernels and Giry (measure-theoretic) kernels; and they provide a rich string-diagram calculus allowing for simple graphical presentations of many calculations and constructions in probability.

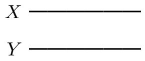
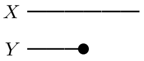
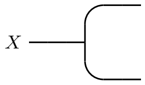
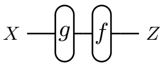
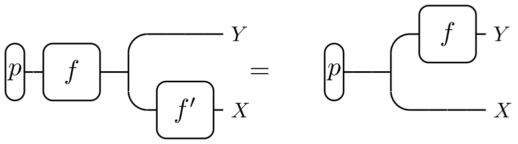
$\mathbb{P}_X(x)\mathbb{P}_Y(y)$ Product distributions	 Parallel composition	$\sum_{y\in Y}\mathbb{P}_{X\times Y}(-,y)$ Marginalisation	 Deletion
$\mathbb{P}_X(x)\delta(x,x')$ Diagonal distributions	 Copying	$\sum_{y\in Y}\mathbb{P}_f(z\mid y)\mathbb{P}_g(y\mid x)$ Chapman-Kolmogorov	 Sequential composition

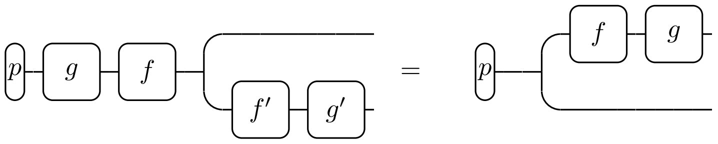
Figure 1 The graphical representation of common formulas for probability distributions.

A categorical translation of Bayes' law allows for a general definition of a *Bayesian inverse* to a morphism in a Markov category. However, in contrast to many other contexts where we have a notion of dualising morphisms, Bayesian inverses depend on an extra piece of data: the prior distribution. This is abstracted in our definition as a *state* on an object X, or a morphism out of the monoidal unit, $I \to X$, which represents a (non-conditional) probability distribution on X. This leads us to the abstract definition of a Bayesian inverse [5]:

▶ **Definition 1** (Bayesian Inversion). *Let $f : X \to Y$ and $p : I \to X$, a kernel $f' : Y \to X$ is called a Bayesian inverse of f at p if the following equation holds:*



This categorical based approach to Bayes' law naturally leads to a question of whether Bayesian inverses compose; that is, can we construct a Bayesian inverse of a composite kernel $f \circ g$ as a composite of the separate Bayesian inverses of f and g? Inspecting the relevant equation



(1)

we can observe that this is indeed true if f' and g' are Bayesian inverse to f and g, but we have to ask for f' not to be inverse to f at p, but rather at $g \circ p$: in some sense the type of this composition is dependent on the kernels that it is being applied to. In other words, assuming a hypothetical function $\mathsf{BayesInv}(f, p)$ which computes Bayesian inverses, the composite could be expressed as

$$\mathsf{BayesInv}(f \circ g, p) = \mathsf{BayesInv}(g, p) \circ \mathsf{BayesInv}(f, g \circ p)$$

Notably this has a similar form to the reverse-mode chain rule for Jacobian matrices that underlies backpropagation, $J^\top_{g \circ f}(x) = J^\top_g(f(x))J^\top_f(x)$. As such we think of the composition rule as a *chain rule for Bayesian updating*. To formalise the compositionality of such a construction we would typically like to promote it to a functor on our Markov category, which we denote in general as $\mathcal{C}$. The fact that Bayesian inverses are indexed by p in this way makes choosing the target of such a a Bayesian inversion functor nontrivial however.

One approach to making Bayesian inversion functorial, taken by Cho and Jacobs [5], is to work instead in a category where the objects X are equipped with a choice of $p : I \to X$: in this case, there is always a canonical choice of inverse morphism, making Bayesian inversion into a "dagger" functor. Although such a setting is still a Markov category, this approach does depart from the operational interpretation of morphisms as stochastic kernels, because the distribution produced is already present in the type of the morphism. So the morphisms have more of a "relational" role.

In this paper we take an alternative approach and instead consider a category where the morphisms are indexed families of kernels. Consequently, we can represent the entire $(I \to X)$-indexed family of Bayesian inverses as a single morphism, resolving the problems faced previously. The morphisms of this category have a very similar structure to a *lens* from database theory and functional programming [7], but, instead of performing deterministic updates to data structures, this performs Bayesian updates to beliefs. We therefore call such pairs *Bayesian lenses*.

This category is however "too big" in the sense that it contains everything whose type matches that of the Bayesian inverse. Rather than a failure of the abstraction, we view this as a feature allowing us to encode not only exact Bayesian inversion, but also approximate updaters and other structures. To pick out the specific lenses corresponding to the actual Bayesian inverse of f we therefore use a functor $\mathcal{C} \to \mathbf{BLens}$, which encodes a choice of Bayesian inverse for each kernel.

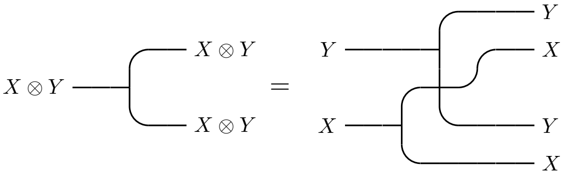
The category of Bayesian lenses is constructed as a fibred category that is closely related to the families fibration, commonly used in the semantics of dependent types. We embrace this relationship by extending **BLens** into a larger category constructed from a generalised families fibration which leads us to a category of *dependent Bayesian lenses*, which have not only indexed families as morphisms, but indexed objects as well. This construction admits Bayesian inverses whose domains are allowed to depend on the prior distribution at which the inverse is taken. In certain Markov categories this allows us to consider inverses as being restricted to the support of the prior, which puts the construction on a neater theoretical foundation, and which clarifies thinking about the Bayesian inversion of structure maps in the category. Since Bayes' law does not define how beliefs should be updated after a zero-probability observation, this falls into the usual pattern in computer science of using a type system to "make illegal states unrepresentable"[1].

[1] This phrase originated with Yaron Minsky in the blog post `https://blog.janestreet.com/effective-ml-revisited/`

2 Preliminaries

▶ **Definition 2** (Markov Categories [8]). *A Markov category is a symmetric monoidal category $\mathcal{C}$ with the structure of a commutative comonoid on each object, such that:*

- *the assignment of counits ("delete" or "discard" maps) to objects is natural; and*
- *comultiplications ("copy" maps) are compatible with the monoidal structure:*



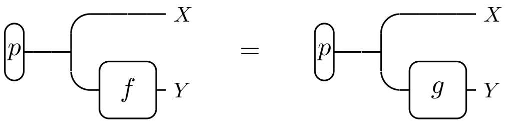
▶ **Example 3** (Stochastic Matrices, [8] example 2.5). There is a Markov category **FinStoch** whose objects are finite sets, and whose morphisms $f : X \to Y$ are stochastic matrices $f : X \times Y \to [0, 1]$, i.e. satisfying $\sum_{y \in Y} f(x, y) = 1$ for all $x \in X$. Identity morphisms are identity matrices, and composition of morphisms is by matrix multiplication, known in this context as the Chapman-Kolmogorov equation, $(g \circ f)(x, z) = \sum_{y \in Y} f(x, y) \cdot g(y, z)$. The symmetric monoidal structure of **FinStoch** is given on objects by cartesian product, and on morphisms by tensor product of stochastic matrices: $(f \otimes g)((x, x'), (y, y')) = f(x, y) \cdot g(x', y')$.

▶ **Example 4** (Gaussian Kernels, [8] section 6). There is a Markov category **Gauss** whose objects are Euclidean spaces $\mathbb{R}^n$, and whose morphisms $f : \mathbb{R}^m \to \mathbb{R}^n$ are triples of a matrix $M \in \mathbb{R}^{m \times n}$, a vector $\mu \in \mathbb{R}^n$ and a positive semidefinite matrix $\sigma \in \mathbb{R}^{n \times n}$, considered to represent an affine function with independent Gaussian noise $f(v) = Mv + \mathcal{N}(\mu, \sigma)$. The definitions of categorical composition and monoidal product are slightly involved, but can be derived from laws of Gaussian probability.

▶ **Example 5** (Probability Kernels, [8] section 4). Giry [10] introduced a monad $\mathcal{G}$ on the category of measurable spaces, now known as the Giry monad, taking each measurable space to the space of probability measures on it, equipped with an appropriate measurable structure. The Kleisli category $\mathrm{Kl}(\mathcal{G})$ is a Markov category, also known as **Stoch**, which allows working with arbitrary measure-theoretic probability and contains many other important Markov categories as subcategories, including the previous two examples. Although this category is in a sense canonical, it suffers from many undesirable properties due to extremely general nature of measure theory. One example which is relevant in the later sections is the spaces representing the support of certain distributions.

Keeping these examples in mind, we refer to the morphisms of such a category as *Markov kernels* or more simply as *kernels*.

▶ **Definition 6** (Almost-Sure Equality [5]). *A pair of kernels $f, g : X \to Y$ are p-almost-surely equal for some $p : I \to X$, if the following equality holds*



In this case we write $f \simeq_p g$.

We note that this equation is similar in shape to that which defines a Bayesian inverse. Indeed this pattern of considering a kernel composed onto one branch of a copy is a standard way to consider a kernel in the context of a distribution on its domain. The appearance of this pattern in the definition of the Bayesian inverse means exactly that we are defining the concept only up to almost-sure equality.

▶ **Proposition 7** (Bayesian inverses are almost-surely equal). *For any kernel $f : X \to Y$ and state $p : I \to X$, if two kernels $g, g' : Y \to X$ both satisfy the conditions of a Bayesian inverse to f, then $g \simeq_{f \circ p} g'$.* ◀

Considering the interpretation of these equations, this result makes intuitive sense. It essentially says that Bayesian inversion is uniquely defined only on the points where Bayes' law would not have you divide by zero. This allows us to be mindful of the partiality of Bayes' law without requiring the added complication of considering partial maps. However, this will complicate matters later when we establish the functoriality of Bayes' law. In fact, as we will see, we will need some extra coherence assumptions to ensure that Bayesian inversion is functorial, rather than only almost-surely functorial.

3 Bayesian Updates Compose Optically

The Bayesian inverse of a kernel is defined with respect to a prior state on the domain of the kernel. We want to consider the general inverse of a kernel $A \to B$ as a function which assigns to each prior $p : I \to A$, a p-inverse $B \to A$. However the space of priors with respect to which a kernel can be inverted depends on the domain of the kernel: it is the set of states $\mathcal{C}(I, \mathrm{dom}(f))$. To formalise this dependence we construct an *indexed category*. Just as indexed sets $(X_i)_{i \in I}$ can be thought of as functions $I \to \mathsf{Set}$, we define indexed categories as (pseudo)functors into **Cat**. We consider Bayesian inverses in the context of an indexed category $\mathcal{C} \to \mathbf{Cat}$ sending X to a category of $\mathcal{C}(I, X)$-indexed kernels.

▶ **Definition 8** (The **Stat** construction). *We define the indexed category of $\mathcal{C}$-state-indexed kernels, $\mathbf{Stat} : \mathcal{C}^{op} \to \mathbf{Cat}$, as follows.*

For each X, $\mathbf{Stat}(X)$ is the category of $\mathcal{C}(I, X)$-indexed kernels, where
- *objects are the objects of $\mathcal{C}$;*
- *morphisms $A \to B$ are functions $\mathcal{C}(I, X) \to \mathcal{C}(A, B)$; and*
- *composition and identities are pointwise given by the corresponding structure in $\mathcal{C}$.*

For $f : X \to Y$ in $\mathcal{C}$ we obtain a reindexing functor $f^* : \mathbf{Stat}(Y) \to \mathbf{Stat}(X)$ which
- *acts as the identity on objects; and*
- *reindexes functions by sending $\sigma : \mathcal{C}(I, Y) \to \mathcal{C}(A, B)$ to $f^*\sigma$ defined by:*

$$\begin{tikzcd} \mathcal{C}(I, X) \arrow[r] & \mathcal{C}(I, Y) \arrow[r] & \mathcal{C}(A, B) \\ p \arrow[r, mapsto] & f \circ p \arrow[r, mapsto] & \sigma(f \circ p) \end{tikzcd}$$

Since the reindexing functors are defined by pre-composition it is easy to verify that **Stat** *is indeed a functor.*

This provides sufficient expressive power to represent general Bayesian inverses. For a kernel $f : X \to Y$, the general Bayesian inverse of f is a morphism $Y \to X$ in $\mathbf{Stat}(X)$. However it is awkward to have to work across a large collection of categories in order to represent the inverses for every kernel. Really we would like to assemble the collection of categories into a single category containing all of the inverses. Fortunately this category is described exactly by the *Grothendieck construction* [2, §8.3]. The Grothendieck construction realises an equivalence between indexed categories and *fibrations* – functors with a property of being suitably projection-like. The core of the construction is a disjoint union of the constituent categories, equipped with morphisms induced by the reindexing functors that connect each of the otherwise disjoint components.

Indexed categories	$F : \mathcal{C} \to \mathbf{Cat}$
Fibrations	$P_F : \left(\coprod_{C \in \mathcal{C}} F(C)\right) \to \mathcal{C}$

We proceed to construct the category of all state-indexed kernels in this way, which we will call the category of *Bayesian lenses.*

▶ **Definition 9** (Bayesian Lenses). *We define the category of* Bayesian lenses *as the Grothendieck construction* $\mathbf{BLens}(\mathcal{C}) = \coprod_{X \in \mathcal{C}} \mathbf{Stat}(X)^{op}$.

Explicitly this is a category whose objects are pairs $\binom{X}{A}$ of objects from $\mathcal{C}$, and whose morphisms are of the form $(f, f^\sharp) : \binom{X}{A} \to \binom{Y}{B}$ where f is a kernel $X \to Y$, and $f^\sharp$ is a family of "backward" kernels, $\mathcal{C}(I, X) \to \mathcal{C}(B, A)$. The composition of two such morphisms

$$\begin{pmatrix} X \\ A \end{pmatrix} \xrightarrow{(f,f^\sharp)} \begin{pmatrix} Y \\ B \end{pmatrix} \xrightarrow{(g,g^\sharp)} \begin{pmatrix} Z \\ C \end{pmatrix}$$

is $(g \circ f, (g \circ f)^\sharp)$ where $(g \circ f)^\sharp$ is a function $\mathcal{C}(I, X) \to \mathcal{C}(C, A)$, sending $p : I \to X$ to the composition

$$C \xrightarrow{g^\sharp(f \circ p)} B \xrightarrow{f^\sharp(p)} A.$$

We call the morphisms of this category *Bayesian lenses.*

▶ Remark 10. Contrary to our general description of the Grothendieck construction, we in fact used the "fibrewise" opposite of **Stat**, taking $\mathbf{Stat}(X)^{\mathrm{op}}$ for each X in $\mathcal{C}$. Ultimately this does not change the space of functions we can represent, but it provides a neater type signature for the morphisms we care about. Namely this means the Bayesian inverse of a kernel will be a morphism $\binom{X}{X} \to \binom{Y}{Y}$, rather than $\binom{X}{Y} \to \binom{Y}{X}$. We think of the appearance of the opposite here as signalling that we are working with a kind of bidirectional process.

Spivak [19] proposes that the fibrewise opposite of a fibration be considered as a generalisation of the lens construction from database theory and functional programming [7]. More clearly, if $\mathcal{C}$ is the category of sets (which is degenerately a Markov category), then $f^\sharp : \mathcal{C}(I, X) \to \mathcal{C}(B, A)$ can be equivalently written $f^\sharp : X \times B \to A$, and we recover the standard definition of a lens. This motivates our use of the term "Bayesian lens" for these morphisms.

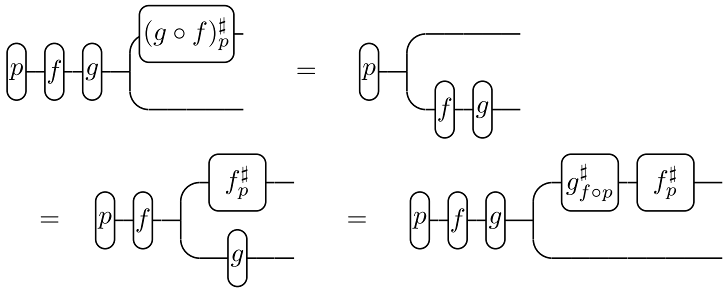
▶ **Proposition 11** (Bayesian inversion is almost functorial). *If $\mathcal{C}$ has Bayesian inverses for ever kernel at every prior, then Bayesian inversion defines a functor $T : \mathcal{C} \to \mathbf{BLens}(\mathcal{C})$ up to almost-sure equality.*

Proof. Because Bayesian inverses are only unique up to almost-equality, T is only almost surely functorial: it maps each object X to $\binom{X}{X}$, and each kernel $f : X \to Y$ to a lens $(f, f^\sharp) : \binom{X}{X} \to \binom{Y}{Y}$ whose inverse component is given by a representative of the almost-surely unique family of Bayesian inversions of f.

Given a pair of composable kernels $X \xrightarrow{f} Y \xrightarrow{g} Z$, we need to check that this mapping is almost surely functorial. Following equation (1), we note that the state with respect to which this functoriality is almost sure is $g \circ f \circ p$, for every state p on X. We therefore need to verify that

$$\begin{pmatrix} X \\ X \end{pmatrix} \xrightarrow{(f,f^\sharp)} \begin{pmatrix} Y \\ Y \end{pmatrix} \xrightarrow{(g,g^\sharp)} \begin{pmatrix} Z \\ Z \end{pmatrix} \quad \simeq_{g\circ f} \quad \begin{pmatrix} X \\ X \end{pmatrix} \xrightarrow{(g\circ f,(g\circ f)^\sharp)} \begin{pmatrix} Z \\ Z \end{pmatrix} .$$

This follows from three applications of Definition 1:



◀

Note that although Markov categories in general may not admit all Bayesian inverses, we can always restrict to a wide subcategory $\mathcal{C}^\dagger$ consisting of only those kernels which admit inverses.

4 Dependent Bayesian Lenses

Those familiar with categorical semantics of type theory might observe that the construction of Bayesian lenses is similar to that of the *families fibration* [12, §1.2]. This construction, commonly used in the interpretation of dependent types, constructs a category whose objects and morphisms are set-indexed families of objects and morphisms from another underlying category. As a fibration this category projects onto the category of sets, by picking out the indexing set or reindexing function under each family.

▶ **Definition 12** (The families fibration [12, §1.2]). *The indexed category of families over a category $\mathcal{C}$ is the functor $\mathbf{Fam}_\mathcal{C} : \mathbf{Set}^{op} \to \mathbf{Cat}$ defined as so:*

- *For a set X, $\mathbf{Fam}_\mathcal{C}(X)$ is the category whose objects are X-indexed families of objects $X \to \mathrm{Ob}(\mathcal{C})$ and whose morphisms $A \to B$ are families of morphisms $\phi : \{x \in X\} \to \mathcal{C}(A_x, B_x)$.*
- *For a function $\alpha : Y \to X$, the induced reindexing functor $\alpha^* : \mathbf{Fam}_\mathcal{C}(X) \to \mathbf{Fam}_\mathcal{C}(Y)$ is given by pre-composition with alpha:*

$$\alpha^*(A) : X \xrightarrow{\alpha} Y \xrightarrow{A} \mathrm{Ob}(\mathcal{C})$$

$$\alpha^*(\phi) : \{y \in Y\} \xrightarrow{\alpha} \{\alpha(x) \in X\} \xrightarrow{\phi} \mathcal{C}\left(A_{\alpha(y)}, B_{\alpha(y)}\right)$$

Alternatively, viewing sets as discrete categories, $\mathbf{Fam}_\mathcal{C}$ is the functor that sends X to the functor category $\mathbf{Cat}(X, \mathcal{C})$ and reindexes by precomposition with the function viewed as a functor between discrete categories.

A crucial difference between this and the indexed category in Definition 8 is that the latter is indexed only by sets of the form $\mathcal{C}(I, X)$ whereas $\mathbf{Fam}_\mathcal{C}(-)$ allows for arbitrary indexing sets. Having the indexing sets take the form $\mathcal{C}(I, -)$ allows us to additionally restrict the reindexing functors to those represented by kernels of $\mathcal{C}$. Formally we can see this as the result of reindexing $\mathbf{Fam}_\mathcal{C}(-)$ itself with the state functor:

▶ **Definition 13** (State-indexed families). *The indexed category* $\mathbf{StFam}_{\mathcal{C}} : \mathcal{C}^{op} \to \mathbf{Cat}$ *is given by the composite of functors:*

$$\mathcal{C}^{op} \xrightarrow{\mathcal{C}(I,-)^{op}} \mathbf{Set}^{op} \xrightarrow{\mathbf{Fam}_{\mathcal{C}}} \mathbf{Cat}$$

$$\begin{tikzcd}
A \arrow[d, "f"'] & \mathcal{C}(I,A) \arrow[d, "f\circ -"] & \mathcal{C}^{\mathcal{C}(I,A)} \\
B & \mathcal{C}(I,B) & \mathcal{C}^{\mathcal{C}(I,B)} \arrow[u, "\alpha\mapsto\alpha(f\circ -)"']
\end{tikzcd}$$

This has as objects $\mathcal{C}(I,X)$*-indexed families of objects from* $\mathcal{C}$ *and as morphisms* $\mathcal{C}(I,X)$*-indexed families of kernels from* $\mathcal{C}$*.*

Note that the categories indexed by $\mathbf{StFam}$ are more general than those indexed by $\mathbf{Stat}$: the objects of $\mathbf{StFam}_{\mathcal{C}}(X)$ are whole functions $\mathcal{C}(I,X) \to \mathrm{Ob}(\mathcal{C})$, whereas an object of $\mathbf{Stat}_{\mathcal{C}}(X)$ is merely a single object of $\mathcal{C}$.

▶ **Proposition 14.** $\mathbf{Stat}_{\mathcal{C}}$ *is a subfunctor of* $\mathbf{StFam}_{\mathcal{C}}$ *given by restricting to subcategories which consist only of constantly-indexed families of objects i.e. those families* $\mathcal{C}(I,X) \to \mathrm{Ob}(\mathcal{C})$ *which are constant functions.* ◀

Observing that $\mathbf{StFam}$ is a generalisation of $\mathbf{Stat}$, we are led to consider whether the fibration constructed from $\mathbf{StFam}$ could profitably be considered as a generalised category of lenses. By analogy with the use of $\mathbf{Fam}$ in dependent type theory we refer to these as *dependent Bayesian lenses.*

▶ **Definition 15** (Dependent Bayesian Lenses). *We define the category of* dependent Bayesian lenses *over* $\mathcal{C}$ *to be the Grothendieck construction of the fibrewise opposite of* $\mathbf{StFam}$*,*

$$\mathbf{DBLens}(\mathcal{C}) = \coprod_{X \in C} \mathbf{StFam}_{\mathcal{C}}(X)^{op} \, .$$

Unpacking this definition we have that:

- Objects of $\mathbf{DBLens}(\mathcal{C})$ are pairs $\binom{X}{A}$ where $X \in \mathcal{C}$ and $A : \mathcal{C}(I,X) \to \mathrm{Ob}(\mathcal{C})$. We think of this as representing a set of dependent pairs, whose elements are $\langle p, a \rangle \in \mathcal{C}(I,X) \times A(p)$.
- Morphisms $\binom{X}{A} \to \binom{Y}{B}$ are pairs $(f, f^\sharp)$ of a kernel $f : X \to Y$ and a family of kernels $f^\sharp : \{p \in \mathcal{C}(I,X)\} \to \mathcal{C}(B(f \circ p), A(p))$.
- The composition is as with the non-dependent version in Definition 9. Aside from the fact that $f^\sharp$ is here considered as a dependent function, the data of a morphism is the same. It is straightforward to verify that the extra dependent typing data still aligns where appropriate.

▶ Remark 16. The functor $\mathcal{C}(I,-)^{\mathrm{op}} : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}^{\mathrm{op}}$ can be made into a lax monoidal functor with structure maps induced by functions $\mathcal{C}(I, X \otimes Y) \to \mathcal{C}(I,X) \times \mathcal{C}(I,Y)$ which map states to pairs of states obtained by deleting either variable. Then it follows by results in [14] that $\mathbf{StFam}$ is the composition of lax monoidal functors, and so its Grothendieck construction inherits a monoidal structure.

5 Support Objects

Giving an abstract account of Bayes' law, Definition 1 promises to be very important in studying Bayesian statistics synthetically, but it is in some way unsatisfying because it only specifies a morphism up to almost-sure equality. For example, considering $\mathbf{FinStoch}$, if a distribution $p : I \to X$ is not fully supported, then the inverse of a kernel $X \to Y$ is only specified by the above definition at the points in the support of p. We can work around this ambiguity however by instead considering inverses as kernels between objects representing the supports of distributions.

Fritz [8] proposes a definition for support objects in a Markov category, but does not develop the idea further. Here we investigate some properties of support objects and explain how they can be used to enhance the theory of abstract Bayesian inversion.

▶ **Definition 17.** *Fix a state $p : I \to X$. An object X_p is called a* support *of p if X_p represents the covariant functor $(\mathcal{C}(X, -)/\simeq_p) : \mathcal{C} \to \mathbf{Set}$.*

This definition succinctly captures the essential properties of the support of a distribution, but it is quite opaque and does not encourage intuition. If an object is a support of another object, intuitively it should behave as a subspace, having an inclusion map which satisfies some extra properties. We state this formally in terms of the existence of certain section-retraction kernels representing this inclusion.

▶ **Proposition 18.** *X_p is a support of $p : I \to X$ if and only if there is a section-retraction pair $X_p \xrightarrow{i} X \xrightarrow{r} X_p$ such that for any kernels $f, g : X \to Y$ we have $f \simeq_p g \iff f \circ i = g \circ i$.*

Proof. Assume X_p is a support. This means we have a natural isomorphism $\Phi : (\mathcal{C}(X, -)/\simeq_p) \to \mathcal{C}(X_p, -)$. We take $i = \Phi_X(\mathrm{id}_X)$, then we see from the following naturality square that the action of Φ must be to pre-compose representative morphisms with i:

$$\begin{tikzcd}
i \arrow[ddd, mapsto] & & & {[\mathrm{id}_X]_{\simeq_p}} \arrow[lll, mapsto] \arrow[ddd, mapsto] \\
& \mathcal{C}(X_p, X) \arrow[d, "{\mathcal{C}(X_p, f)}"'] & \mathcal{C}(X, X)/\simeq_p \arrow[l, "\Phi_X"'] \arrow[d, "{\mathcal{C}(X, f)/\simeq_p}"] & \\
& \mathcal{C}(X_p, Y) & \mathcal{C}(X, Y)/\simeq_p \arrow[l, "\Phi_Y"] & \\
f \circ i & & & {[f]_{\simeq_p}} \arrow[lll, mapsto]
\end{tikzcd}$$

Hence we establish the property that pre-composition by i is an isomorphism between kernels from X and $\simeq_p$-equivalences classes of kernels from X_p. We further have that $\mathrm{id}_{X_p} = \Phi(\Phi^{-1}(\mathrm{id}_{X_p})) = \Phi^{-1}(\mathrm{id}_{X_p}) \circ i$, so we can take the retract to be $r = \Phi^{-1}(\mathrm{id})$.

Conversely, given such an i and r, it is clear that pre-composition by i defines a function $\mathcal{C}(X, Y) \to \mathcal{C}(X_p, Y)$ natural in Y and the assumed property of i guarantees that this is a bijection from $\simeq_p$-equivalence classes. Finally we have that $i \circ r \circ i = i$, so $i \circ r \simeq_p \mathrm{id}_{X_p}$. Hence pre-composition by r is an inverse to $(-) \circ i : (\mathcal{C}(X, -)/\simeq_p) \to \mathcal{C}(X_p, -)$. ◀

While support objects are not necessarily unique, they must be unique up to isomorphism, since two support objects for the same distribution must by definition represent the same presheaf. When we disuss a given support object we therefore really mean a given support object along with a choice of section and retraction.

Now, if we have a distribution on X, $p : I \to X$, and a kernel $f : X \to Y$ we can push p forward to a distribution $f \circ p$ on Y. So f restricts to a kernel $f|_p = r \circ f \circ i : X_p \to Y_{f \circ p}$. Dually we have an inclusion of kernels $g : X_p \to Y_q$ into $\langle g \rangle = i \circ g \circ r : X \to Y$. Using these there is an obvious adjustment to the definition of Bayesian inversion in order to capture Bayesian inverses between supports:

▶ **Definition 19** (Bayesian-inverse-with-support). *Fix kernels $p : I \to X$ and $f : X \to Y$ with support objects X_p and $Y_{f \circ p}$. We call a kernel $f^\sharp_p : Y_{f \circ p} \to X_p$ a* Bayesian inverse with support *if $\langle f^\sharp_p \rangle$ is an ordinary Bayesian inverse for f at p.*

This is very similar to the previous definition. However the presence of support objects in $f^\sharp_p$'s domain and codomain mean that the kernel is more canonical than an ordinary Bayesian inverse. In fact for any ordinary inverse $f^\sharp_p$, its restriction to the distribution $f \circ p$ is an inverse-with-support.

▶ **Theorem 20** (Unique Bayesian Inversion). *Fix kernels $f : X \to Y$ and $p : I \to X$, and support objects X_p and $Y_{f \circ p}$. If f has an ordinary Bayesian inverse at p, then there is a unique inverse-with-support to f at p.*

Proof. See Appendix A. ◀

This final result suggests that we can now define a functor that picks out the canonical inverse for a given kernel. Recall that in Proposition 11 we were almost able to show that Bayesian inversion defines a functor $\mathcal{C} \to \mathbf{BLens}(\mathcal{C})$ except that the functoriality constraint only holds up to almost-equality. Using support objects we improve this into strict functoriality between kernels. Bayesian inversion with supports departs from the previous situation slightly in that the domain and codomain of the inverse kernels vary as the indexing distribution varies. Fortunately this is exactly the extra level of generality afforded us by dependent Bayesian lenses.

▶ **Proposition 21.** *If $\mathcal{C}$ has Bayesian inverses for every kernel, and support objects for every distribution $I \to X$ then the fibred category $\mathbf{DBLens}(\mathcal{C})$ has a section $T : \mathcal{C} \to \mathbf{DBLens}(\mathcal{C})$ sending kernels to families of their Bayesian inverses between support objects.*

Proof. We write explicitly the mapping defining T. On objects $T(X) = \binom{X}{S_X}$ where $S_X : \mathcal{C}(I, X) \to \mathrm{Ob}(\mathcal{C})$ maps p to a choice of support object X_p. On kernels $f : X \to Y$, we have $T(X) = (f, f^\sharp) : \binom{X}{S_X} \to \binom{Y}{S_Y}$ where for each $p \in \mathcal{C}(I, X)$, $f^\sharp_p : Y_{f \circ p} \to X_p$ is given by the Bayesian inverse with support of f at p.

The functoriality of T follows similarly to in Proposition 11. Briefly, if f and g are composable Bayesian inverses with support, then $\langle f \rangle$ and $\langle g \rangle$ are composable ordinary inverses, so by Proposition 11 their composition must be an inverse. By the definition of the inclusions it is immediately that $\langle f \rangle \circ \langle g \rangle = \langle f \circ g \rangle$, so $f \circ g$ is an inverse-with-support. Then by uniqueness, the fact that the composition $f \circ g$ is a Bayesian inverse implies the strict equality required for functoriality. ◀

▶ Remark 22. Proposition 21 proves the compatibility of sequential composition in a Markov category with Bayesian inversion, however there is another axis for composition in our graphical notation. This is the parallel composition, or monoidal product of $\mathcal{C}$. To ask for this structure to be compatible with inversion is to ask for the functor T to respect the monoidal structure. Indeed as we noted in Remark 16, we may naturally define a monoidal product on $\mathbf{DBLens}(\mathcal{C})$, then T straightforwardly inherits the structure of a *lax monoidal functor* with respect to this.

▶ Remark 23. Most of the results developed in this section relied on the existence of support objects. Indeed this is satisfied for our first two examples Example 3 and Example 4, but this is quite a strong assumption in general. The approach to functorial Bayesian inversion taken in [5] and later expanded on in [8] is to work in a category $\mathbf{ProbStoch}(\mathcal{C})$ whose objects are equipped with a choice of distribution and whose morphisms are quotiented by almost-sure equality with respect to these chosen distributions. This makes exact functoriality straightforward to prove, but it detracts from the interpretation of morphisms as stochastic functions. Working in $\mathbf{ProbStoch}(\mathcal{C})$ we are unable to think of Bayesian updating as a dynamic process which maps observations to new distributions, since the resultant distribution is already encoded in the codomain.

We could instead take a hybrid approach, using dependent Bayesian lenses where the forwards kernel is still valued in $\mathcal{C}$ but the "backwards' mapping is in $\mathbf{ProbStoch}(\mathcal{C})$. This means the type of a general inverse can become a function $\{p \in \mathcal{C}(I, X)\} \to \mathbf{ProbStoch}(\mathcal{C})((Y, f \circ p), (X, p))$. This provides a lot of the same theoretical niceties of support objects, including exact functoriality, while still applying in a more general context. We leave the further investigation of this construction to future work.

6 Example: Estimating Transition Probabilities in Markov Chains

To illustrate the compositional nature of Bayesian inversion in practice we consider how to view a typical Bayesian inference problem in this framework. Namely we use as our example, the common problem of learning transition probabilities for a Markov chain from an observed sequence of states [21].

In the category **FinStoch**, a Markov chain is nothing but an endomorphism $t : S \to S$ for some finite set S of *states*, with an initial state distribution $s : I \to S$. Often we do not know the actual transition probabilities, but instead have a family of distributions dependent on another external parameter, $t : S \otimes \Theta \to S$. In such a situation we want to choose an optimal value of Θ based on a sequence of observations of states in the past. A common technique for choosing this value is to consider the observed state sequence as an element of the space $S \otimes \ldots \otimes S$, with which we can perform a Bayesian update. That is, given a state sequence $(s_1, \ldots, s_n)$ we would like to compute a posterior distribution $\mathbb{P}(\Theta | s_1, \ldots, s_n)$.

From the data involved we can define a family of maps $f_n : \Theta \to S^{\otimes n}$ describing the distribution over n-length state sequences corresponding to any value of Θ (see Figure 2 for an example). Computing the Bayesian inverse of f_n we obtain kernels which map sequences $S^{\otimes n}$ to posterior distributions over Θ. Of course, as we have already discussed, in order to compute this all we need is the Bayesian inverse for the transition matrix t. Knowing the functorial relationship between "forward" models f_n and their corresponding inverse models, we can build an inverse for f_n out of many copies of the inverse for t.

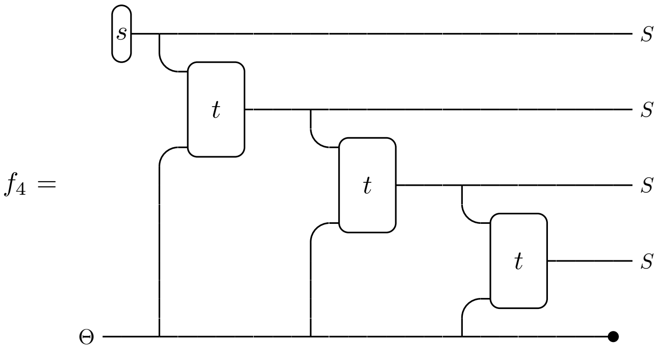


Figure 2 An example of a "state-trace" kernel obtained from a Markov chain, yielding state sequences of length 4.

Figure 2 depicts a typical composite which we may want to invert. It consists only of four kinds of non-trivial cell: the transition matrix t, the initial state distribution s, copy maps $\mathsf{copy}_S : S \to S \otimes S$ and $\mathsf{copy}_\Theta : \Theta \to \Theta \otimes \Theta$, and a delete map $\mathsf{delete}_\Theta : \Theta \to I$. As t is an arbitrary kernel there is little we can say about its Bayesian inverse, but we can fully characterise the inverses for the other kinds of cells. The inverse of s at p should have a type $s^\sharp_p : S_p \to I$, but by the uniqueness of the delete map this is equal to $\mathsf{delete}_{S_p} : S_p \to I$. Dually, filling in the definition of Bayesian inversion for delete_Θ at p requires that $(\mathsf{delete}_\Theta)^\sharp_p$ is exactly $p : I \to \Theta$.

Finally we consider the Bayesian inverse of copy_S. Of course the case for copy_Θ is identical. In the case of non-dependent Bayesian lenses, without supports, it is difficult to understand what these copy maps do. The copy operation is supported on only a small subset of its domain. Intuitively this is the diagonal set $\{(s, s)|s \in S\}$, but in the abstract case this set notation is not valid. We may wonder if there is a way that we can describe this using only the axiomatic setting we have been working with so far. Indeed, we can show that there is in fact an isomorphism of the support object $(S \otimes S)_{\mathsf{copy}_\Theta \circ p} \cong S_p$ and moreover, the morphism witnessing this isomorphism is exactly the Bayesian inverse of copy.

This isomorphism is a very powerful fact. It enforces in the type of a morphism, using the notion of dependent typing present in $\mathbf{DBLens}(\mathcal{C})$, that the observations we make from a process involving copying should preserve the exact equalities expected given our knowledge about the generative processes from which the observations originated.

It is not difficult to derive an explicit formula for this with traditional probability theory, but our structural viewpoint can offer a more pedagogical approach to describing inference algorithms. Using a 2-dimensional syntax for kernels and understanding the compositional nature of Bayesian inversion, a lot of the apparent complexity of equations in statistics is clarified by applying operations piecewise on string diagrams. It additionally becomes straightforward to see how to extend this to more complicated scenarios. For example, a hidden Markov model is obtained from the above example simply by postcomposing each wire with an additional kernel $o : S \to O$. Then compositionality described exactly how to extend a solution to account for the additional mapping.

This approach provides not just additional pedagogy, but also is very amenable to algorithmic study: as future work, we hope to investigate ways in which, by isolating the basic repeated units and their compositional structure, we can automatically generate a lot of the additional data required and can perhaps facilitate certain optimisations.

7 Further work

Our work situates Bayesian inversion amongst a range of bidirectional processes observed in different contexts, each of which exhibit lens (or lens-like) structure and "cybernetic" application [4]: reverse-mode automatic differentiation (generalizing backpropagation of error) [6]; economic games [9]; reinforcement learning [11]; and database updating [7]. It was in this latter setting, in the context of functional programming, that lenses were first described, and generalizations of lenses remain in popular use in that setting. This points to a first avenue for future work: a new approach to probabilistic programming that incorporates both Bayesian and differential updating, extending the currently popular use of deterministic lenses, and with general application to cybernetic systems.

Probabilistic programming languages allow the programmer essentially to construct Markov kernels in a compositional way using a standard programming language syntax, and perform inference on the resulting models [20]. Typically, performing inference is not transparently compositional, and so we hope that our results will lead to improvements here, for instance by allowing the programmer to "amortize" parts of an inference process. Perhaps more importantly, our approach could lead to the use of a dependent type system to statically rule out events of measure zero. Furthermore, in a generically bidirectional language of the kind we envisage, we expect that compilers will be able to share optimizations across differential and probabilistic parts.

We expect these ideas not only to be of use to language designers therefore, but also to users. In many applications of probabilistic inference, one first constructs a "joint model": a joint state across all the variables of interest, which may factorise according to a Bayesian network or other graphical model. Because computing exact Bayesian inversions involves marginalization (normalization: the sum or integral in the Chapman-Kolmogorov equation), and such a computation is often intractable, one usually resorts to approximate methods, and this is where the "extra" morphisms in categories of Bayesian lenses come in: they can be understood as approxmate inverses.

By parameterizing these lenses, one can associate to them *loss functions* that characterize "how far" a given inversion is from optimality[2]; like the inversions themselves, these losses depend on both the priors and the observations, and they again compose according to a lens pattern. The third author, in the unpublished works [17, 18], has begun developing this programme, based on ideas from compositional game theory [9]. It results in an account of approximate inference whose algorithms are "correct by construction", which may be an advantage over traditional methods which simply start from a given joint distribution. Moreover, because the loss functions are "local" to each lens, the resulting framework captures the compositional structure seemingly exhibited by "predictive coding" neural circuits in the brain [1]. Similarly, by making use of the lens structure presented here, the framework suggests a formal unification of backprop and predictive coding, as sought by various authors [13, 15], and it reveals connections to the method of backward induction in reinforcement learning [11]. We hope that future developments make use of these relationships, so that we may build intelligent systems that are both efficient and well-understood.

References

1 Andre M. Bastos, W. Martin Usrey, Rick A. Adams, George R. Mangun, Pascal Fries, and Karl J. Friston. Canonical microcircuits for predictive coding. *Neuron*, 2012. doi:10.1016/j.neuron.2012.10.038.

2 Francis Borceux. *Handbook of Categorical Algebra 2. Categories and Structures.* Cambridge University Press, 1994. doi:10.1017/CBO9780511525865.

3 Dylan Braithwaite and Jules Hedges. Dependent Bayesian Lenses: Categories of Bidirectional Markov Kernels with Canonical Bayesian Inversion, 2022. arXiv:2209.14728.

4 Matteo Capucci, Bruno Gavranović, Jules Hedges, and Eigil Fjeldgren Rischel. Towards Foundations of Categorical Cybernetics. *ACT21*, 2022. doi:10.4204/EPTCS.372.17.

5 Kenta Cho and Bart Jacobs. Disintegration and Bayesian inversion via string diagrams. *MSCS*, 2019. doi:10.1017/S0960129518000488.

6 Geoffrey S. H. Cruttwell, Bruno Gavranović, Neil Ghani, Paul Wilson, and Fabio Zanasi. Categorical foundations of gradient-based learning. *ESOP22*, 2022. doi:10.1007/978-3-030-99336-8_1.

[2] Examples of such loss functions include the relative entropy (Kullback-Leibler divergence) and variational free energy ("evidence lower bound").

7 J. Nathan Foster, Michael B. Greenwald, Jonathan T. Moore, Benjamin C. Pierce, and Alan Schmitt. Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem. *TOPLAS*, 2007. doi:10.1145/1232420.1232424.

8 Tobias Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 2020. doi:10.1016/j.aim.2020.107239.

9 Neil Ghani, Jules Hedges, Viktor Winschel, and Philipp Zahn. Compositional game theory. *LICS18*, 2018. doi:10.1145/3209108.3209165.

10 Michèlle Giry. A categorical approach to probability theory. *Categorical aspects of Topology and Analysis*, 1982. doi:10.1007/BFb0092872.

11 Jules Hedges and Riu Rodríguez Sakamoto. Value Iteration Is Optic Composition. *ACT22*, 2023. Forthcoming.

12 Bart Jacobs. *Categorical Logic and Type Theory*. Elsevier Science, 1999.

13 Beren Millidge, Tommaso Salvatori, Yuhang Song, Rafal Bogacz, and Thomas Lukasiewicz. Predictive coding: Towards a future of deep learning beyond backpropagation? *IJCAI22*, 2022. doi:10.24963/ijcai.2022/774.

14 Joe Moeller and Christina Vasilakopoulou. Monoidal Grothendieck construction, 2020.

15 Robert Rosenbaum. On the relationship between predictive coding and backpropagation. *PLoS ONE*, 2022. doi:10.1371/journal.pone.0266102.

16 Toby St. Clere Smithe. Bayesian updates compose optically, 2020. arXiv:2006.01631.

17 Toby St. Clere Smithe. Compositional Active Inference I: Bayesian Lenses. Statistical Games, 2022. arXiv:2109.04461.

18 Toby St Clere Smithe. *Mathematical Foundations for a Compositional Account of the Bayesian Brain*. PhD thesis, University of Oxford, 2023. arXiv:2212.12538.

19 David I. Spivak. Generalized Lens Categories via functors $\mathcal{C}^{\mathrm{op}} \to \mathsf{Cat}$, 2019. arXiv:2109.04461.

20 Jan-Willem van de Meent, Brooks Paige, Hongseok Yang, and Frank Wood. An Introduction to Probabilistic Programming, 2021. arXiv:1809.10756.

21 H. Wu and F. Noé. Maximum a posteriori estimation for markov chains based on gaussian markov random fields. *ICCS10*, 2010. doi:10.1016/j.procs.2010.04.186.

A Uniqueness of Bayesian Inversion

▶ **Notation 24.** *In the graphical language we denote the inclusion and restriction kernels for a support object respectively as follows:* X_p ─<├─ X X ─┤>─ X_p

This is a convenient shorthand for indicating the section-retract relation between sections and restrictions, however care must be taken when calculating with these. Since we are not labelling the triangles, we should only pair inclusions and restrictions corresponding to the same states; the characterising equations only hold for such pairs. However in practice it is difficult to naturally arrive at a situation where this caveat can be an issue, so this is less of a shortcoming that it might seem.

We have strict equality of inclusions followed by restrictions, but in the converse we have almost-equality of restrictions followed by inclusions.

▶ **Lemma 25.** *Restriction is almost inverse to inclusion:*

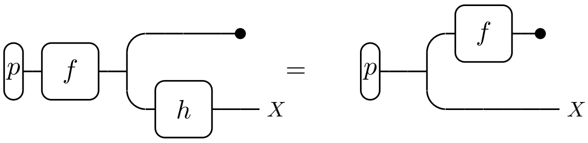
$$X \xrightarrow{r} X_p \xrightarrow{i} X \quad \simeq_p \quad X \xrightarrow{id} X.$$

Proof. Note that $i \circ r \circ i = i$ so by the quotienting property of $(-) \circ i$ we have $i \circ r \simeq_p \mathrm{id}$. ◀

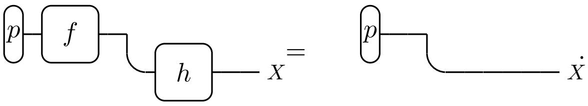
▶ **Proposition 26.** *Fix kernels $f : X \to Y$ and $p : I \to X$, and support objects X_p and Y_{fp}. Then inverses-with-support of f at p are in bijection with $\simeq_{fp}$-equivalence classes of ordinary Bayesian inverses.*

Proof. We first exhibit a map Ψ from inverses-with-support to ordinary inverses. Let $g : Y_{fp} \to X_p$ be a Bayesian inverse with support of f at p. By the definition of inverses with support this means that $\Psi(g) := \langle g \rangle$ is an ordinary Bayesian inverse.

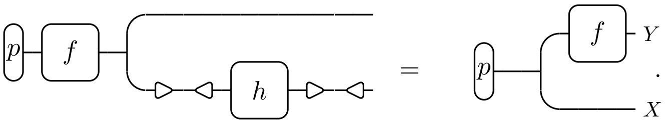
Conversely if $h : Y \to X$ is an ordinary Bayesian inverse to f at p, we want to define an inverse with support $\tilde{\Psi}(h) : Y_{fp} \to X_p$. As a first guess we might try the restriction $h|_{fp}$ but we see that the codomain of this kernel does not match. Namely we have $h|_{fp} : Y_{fp} \to X_{hfp}$ but require a codomain of X_p. In fact we can verify that $hfp = p$: since h is a Bayesian inverse of f at p we have that



and so by the naturality of the delete map:



So it is well defined to set $\tilde{\Psi}(h) := h|_{fp}$. We can see that this is an inverse-with-support: to show this we must show that $\langle h|_{fp} \rangle$ is an ordinary Bayesian inverse. i.e. that



But this is follows straightforwardly by two applications of Lemma 25 and the fact that h is a Bayesian inverse.

We finally have that $\Psi(-)$ is inverse to $\tilde{\Psi}$ when viewed as maps to/from equivalence classes:

$$\begin{aligned}\Psi(\tilde{\Psi}(h)) &= i_p \circ r_p \circ h \circ i_{fp} \circ r_{fp} \\ &\simeq_{fp} i_p \circ r_p \circ h \\ &\simeq_{fp} h\end{aligned}$$

where the final equivalence uses the fact that h is a Bayesian inverse to move it out of the way, similarly to the previous chain of equalities. ◀

Noting that all Bayesian inverses to f at p must be $(f \circ p)$-almost equal we obtain Theorem 20 as a corollary, that Bayesian inverses with support are unique.

▶ **Theorem 20** (Unique Bayesian Inversion). *Fix kernels $f : X \to Y$ and $p : I \to X$, and support objects X_p and $Y_{f \circ p}$. If f has an ordinary Bayesian inverse at p, then there is a unique inverse-with-support to f at p.*