

Compositionality in algorithms for smoothing

Moritz Schauer¹, Frank van der Meulen², and Andi Q. Wang³

¹Chalmers University of Technology and University of Gothenburg, Sweden

²Department of Mathematics, Vrije Universiteit Amsterdam, The Netherlands

³Department of Statistics, University of Warwick, UK

Backward Filtering Forward Guiding (BFFG) is a bidirectional algorithm used for Bayesian inference on partially observed systems, first proposed in Mider et al. [2021] and further studied in Van der Meulen and Schauer [2022]. In category theory, optics have been proposed for modelling systems with bidirectional data flow. We connect BFFG with optics by demonstrating that the forward and backwards map together define a functor from a category of Markov kernels into a category of optics, which is furthermore lax monoidal in the case when the guiding functions and kernels used in the backward step coincide with the generative dynamics.

1 Introduction

With recent advances in computing power and hardware architectures, statistical and machine-learning models and algorithms have increased in scale and complexity. A key technique to understand and analyse complex algorithms is to understand their compositional structure: what are the fundamental building blocks that constitute an algorithm, and how are they related?

Mathematically, the study of structure and composition is the essence of *category theory* Pierce [1991]. As such, the past decade has witnessed an increased application of category theory to machine learning, e.g. Fritz et al. [2024], Shiebler et al. [2021], Fong et al. [2019]. One key underlying idea is to focus on compositionality and abstract away specific implementational details.

In this work, we will make use of tools from category theory in order to study an algorithm for Bayesian inference on partially observed systems. Such systems include *state space models*, also known as hidden Markov models, which consist of a latent Markov process that is only partially observed, possibly subject to noise. A visualisation as a directed graph is given in Figure 1; \bullet and \circ correspond to latent and observed vertices respectively. With X_s denoting the value at vertex s , the values at the leaf vertices are generated as follows:

$$\begin{aligned} X_{t_0} | X_r &\sim \kappa_{r,t_0}(X_r, \cdot); \\ X_{t_i} | X_{t_{i-1}} &\sim \kappa_{t_{i-1},t_i}(X_{t_{i-1}}, \cdot), \quad 1 \leq i \leq n; \\ X_{v_i} | X_{t_i} &\sim \kappa_{v_i,t_i}(X_{t_i}, \cdot). \end{aligned}$$

Here, the $(\kappa_{s,t})$ are Markov kernels (Cf. Section 2.1) that represent the conditional probability distributions by which the process evolves on the directed graph. State-space models have widespread use in many fields including engineering, economics and biology (see e.g. Rabiner and Juang [2007], West and Harrison [1997] and Durbin and Koopman [2012]).

Moritz Schauer: smoritz@chalmers.se, <https://math.chalmers.se/~smoritz/>,  0000-0003-3310-7915

Frank van der Meulen: f.h.van.der.meulen@vu.nl, <https://fmeulen.github.io/>,  0000-0001-7246-8612

Andi Q. Wang: andi.wang@warwick.ac.uk, <https://warwick.ac.uk/fac/sci/statistics/staff/academic-research/wang/>,  0000-0001-9551-6592

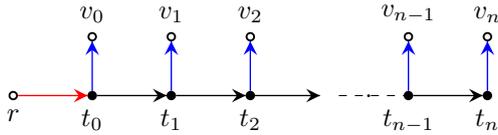


Figure 1: Visualisation of a state-space-model. The process starts from the known root vertex r and evolves over times t_0, t_1, \dots, t_n . At each time t_i , a partial observation at vertex v_i is assumed.

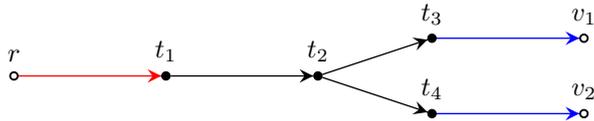


Figure 2: Example of a stochastic process on a directed tree. The root-node is denoted by r which, without loss of generality, is assumed to be known. We assume at each vertex sits a random quantity, and only at the leaf vertices v_1 and v_2 a realisation of that random quantity is observed.

In this paper we consider the more general setup where X is a stochastic process on a directed tree. The process is assumed to be observed exactly at its leaf vertices. A simple example is given in Figure 2.

The smoothing problem consists of inferring the distribution of the process at all interior (non-observed) vertices, conditional on the observations at the leaves. For the example in Figure 2 this amounts to $\{X_{t_i}, 1 \leq i \leq 4\}$. Cappé et al. [2005] present multiple approaches towards this problem for the specific setting of state-space models: (i) normalised forward-backward recursion; (ii) forward decomposition; (iii) backward decomposition. The computations involved can only be performed in closed-form in very specific settings, such as when the latent states take values only in a finite set. The celebrated Kalman filter-smoother for linear Gaussian systems is another well-known example (Bishop [2007]).

Mider et al. [2021] introduced the *Backward Filtering Forward Guiding (BFFG)* algorithm (Algorithm 2), which uses the forward decomposition with simplified dynamics in the backward filtering pass for X . BFFG provides a general framework for dealing with the smoothing problem, encompassing both the Kalman and finite-state space settings as special cases. When no separate simplifying dynamics are used, the algorithm reduces to *Backward Filtering Forward Sampling (BFFS)*, Algorithm 1. The well-known *Forward Filtering Backward Sampling (FFBS)* algorithm for inference in state space models (Frühwirth-Schnatter [1994], Carter and Kohn [1994]) can be seen as BFFS applied to the *time-reversed* process.

The BFFG algorithm is characterised by a *bidirectional data flow*: there is both a forward and backward pass. Within category theory, *lenses* and *optics* have been introduced as abstract models for bidirectionality; see Riley [2018]. It is the aim of this paper to provide a categorical perspective on the BFFG algorithm using optics. In particular, in Theorems 8 and 11 we provide compositionality results for smoothing. More precisely, we show that the BFFG algorithm defines a functor from a category of Markov kernels into a category of optics, which in the case of BFFS is *lax monoidal*. In other words, the optic obtained from parallel or sequential composition of Markov kernels is equivalent to the corresponding composition of optics of the separate Markov kernels. Since FFBS (Frühwirth-Schnatter [1994]) can be seen as a special case of BFFS, analogous compositionality results for FFBS are a direct consequence of our results. While BFFG in general fails to be lax monoidal, this does not typically affect its practical application in Monte Carlo inference, see Remark 17.

This categorical perspective enables prospective users to break down this complex algorithm into its constituent parts, to aid implementation and comprehensibility. It demonstrates that there is a higher-order structure-preserving core within such smoothing algorithms. In particular, our results show that it suffices to define forward and backward maps over each edge in the graph (as in

Definitions 8 and 9 in Section 4). Upon pairing both maps on an edge via the optic, the smoothing algorithm on a directed tree then consists of a composition of optics. Several illustrations of this are discussed in Section 5.

1.1 Related work

There have been a number of recent papers seeking to utilise concepts from category theory to study, understand and develop algorithms arising in computational statistics and machine learning. To this end, the development of *categorical probability* has been very fruitful; see for instance, Fritz [2020], Cho and Jacobs [2019]. This approach sees Markov kernels and their composability via Chapman–Kolmogorov as primary, and enables the use of powerful tools from category theory such as string diagrams. In this work, we will use the fact that Markov kernels form a category, recalled in Proposition 4.

A notable recent work is Fritz et al. [2024] (see also the parallel work of Virgo [2023]), which uses the framework of Markov categories to abstract and generalize methods for inference on hidden Markov models (HMMs). This enables the authors to provide a unified treatment of several such smoothing algorithms. In contrast with the present work, we are less interested in studying algorithms for HMMs in the abstract, but focus on the Backward Filtering Forward Guiding algorithm in particular – which can be applied more generally than the HMM setting – and in fact show that this algorithm itself possesses a compositional structure.

Optics as models of bidirectional data flow have also found numerous applied uses. For instance, in the recent works of Smithe [2020, 2021], Braithwaite et al. [2023], a category of optics termed *Bayesian lenses* is used to give a compositional account of Bayesian inference itself. Further detailed comparisons with these particular works are given in Section 4. Optics have also been applied in various other places Gavranović [2022], such as the Haskell library *lens* and in the study of reverse-mode automatic differentiation Vertechni [2022].

1.2 Outline

In Section 2 we recap some results on Markov kernels and present the BFFG algorithm. In Section 3 we recall the necessary definitions from category theory in order to state our results; specific attention is paid to the category of optics. The categorical results on BFFG together with the main results of the paper, Theorem 8 and Theorem 11, are in Section 4. Section 5 concludes the paper with several examples.

Acknowledgment: We thank Keno Fischer, Philipp Gabler, Evan Patterson, Chad Scherrer and Bruno Gavranović for helpful discussions. AQW was partially supported by a Discipline Hopping Award from the Prob_AI Hub (EP/Y028783/1). We are very grateful to the associate editor and anonymous referees for their extremely helpful comments, including the suggestion of the first/second terminology.

2 Backward Filtering Forward Guiding

In this section, we describe the BFFG algorithm introduced in Mider et al. [2021] and generalised in Van der Meulen and Schauer [2022]; see also Van der Meulen [2022] for a non-technical introduction. We first recap standard results on Markov kernels.

2.1 Markov kernels

Let $S = (E, \mathfrak{B})$ and $S' = (E', \mathfrak{B}')$ be measurable spaces. A Markov kernel from source S to target S' is denoted by $\kappa: S \rightarrow S'$ (note the specific arrow). That is, $\kappa: E \times \mathfrak{B}' \rightarrow [0, 1]$, where (i) for

fixed $B \in \mathfrak{B}'$ the map $x \mapsto \kappa(x, B)$ is \mathfrak{B} -measurable and (ii) for fixed $x \in E$, the map $B \mapsto \kappa(x, B)$ is a probability measure on S' .

On a measurable space S , denote by $\mathcal{M}(S)$ and $\mathbf{B}(S)$ the sets of nonnegative measures and bounded measurable functions on S respectively. The kernel κ induces a *pushforward* of a measure μ on S to a measure on S' via

$$\mu\kappa(\cdot) = \int_{E'} \kappa(x, \cdot)\mu(dx), \quad \mu \in \mathcal{M}(S). \quad (2.1)$$

We can define a linear operator, continuous with respect to the topologies generated by the supremum norm, $\kappa: \mathbf{B}(S') \rightarrow \mathbf{B}(S)$ given by

$$\kappa h(\cdot) = \int_E h(y)\kappa(\cdot, dy), \quad h \in \mathbf{B}(S'). \quad (2.2)$$

We will refer to this operation as the *pullback* of h under the kernel κ . Markov kernels $\kappa_1: S_0 \rightarrow S_1$ and $\kappa_2: S_1 \rightarrow S_2$ can be composed by the Chapman–Kolmogorov equations to obtain a kernel $S_0 \rightarrow S_2$, this (associative) composition is here written as the juxtaposition

$$\kappa_2 \circ \kappa_1(x_0, \cdot) = (\kappa_1\kappa_2)(x_0, \cdot) = \int_{E_1} \kappa_2(x_1, \cdot)\kappa_1(x_0, dx_1), \quad x_0 \in E_0. \quad (2.3)$$

We can furthermore define for $\kappa: S_1 \rightarrow S_2$, $\kappa': S'_1 \rightarrow S'_2$ the product kernel $\kappa \otimes \kappa': S_1 \otimes S'_1 \rightarrow S_2 \otimes S'_2$ on the cylinder sets of $S_2 \otimes S'_2$ by

$$(\kappa \otimes \kappa')((x, x'), B \times B') = \kappa(x, B)\kappa'(x', B'), \quad (2.4)$$

where $x \in E_1$, $x' \in E'_1$, $B \in \mathfrak{B}_2$, $B' \in \mathfrak{B}'_2$, and then extend it to a kernel on the product measure space.

2.2 Description of the stochastic dynamics on a tree

Using Markov kernels, we can define a “computational graph” of the stochastic process evolving on a *directed tree*. This is most easily done through an example. Consider the tree depicted in Figure 2. Though this visualisation is popular in the literature on Bayesian networks – highlighting vertices at which random quantities “sit” – from a categorical point of view it is more natural to convert this figure into the *string diagram* given in Figure 3; for other papers also representing Bayesian networks using string diagrams, see Fong [2012], Jacobs et al. [2019], Fritz and Klingler [2023], Fritz and Liang [2023]. The string diagram corresponding to a state-space model is shown in Figure 4, where for simplicity we took $n = 2$.

Here, for any measurable space S , the copy kernel $\triangleleft: S \rightarrow S \otimes S$ is a Markov kernel defined by

$$\triangleleft(x, dy) = \delta_x(dy_1)\delta_x(dy_2), \quad y = (y_1, y_2). \quad (2.5)$$

This Markov kernel encodes the process of making a copy of x when the process branches into conditionally independent sub-branches. The “computational graph” of the dynamics along the tree is then given by

$$\kappa_{r,t_1}\kappa_{t_1,t_2}\triangleleft(\kappa_{t_2,t_3} \otimes \kappa_{t_2,t_4})(\kappa_{t_3,v_1} \otimes \kappa_{t_4,v_2}).$$

By considering such generations, we see that the sequential structure of this computational graph can be simplified to $\kappa_1\kappa_2\kappa_3\kappa_4\kappa_5$, where $\kappa_1 = \kappa_{r,t_1}$, $\kappa_2 = \kappa_{t_1,t_2}$, $\kappa_3 = \triangleleft$, $\kappa_4 = \kappa_{t_2,t_3} \otimes \kappa_{t_2,t_4}$ and $\kappa_5 = \kappa_{t_3,v_1} \otimes \kappa_{t_4,v_2}$. More generally, for a directed tree, we sequentially order the vertices, by partitioning the vertices in *generations* $\Gamma_1, \Gamma_2, \dots, \Gamma_n$ and $\Gamma_0 = \{0\}$, $\Gamma_{n+1} = \mathcal{V}$, such that all parents of vertices $s \in \Gamma_i$ are in $\bigcup_{0 \leq j < i} \Gamma_j$. Write $X_i = X_{\Gamma_i}$ and set

$$\kappa_i(x_{i-1}; dx_i) = \mathbb{P}(X_i \in dx_i \mid X_{i-1} = x_{i-1}).$$

Therefore, in the following, we assume a stochastic process on a line-graph, with vertices denoted by $\{0, 1, \dots, n+1\}$, where the root-vertex 0 is assumed to be known and the transition from vertex

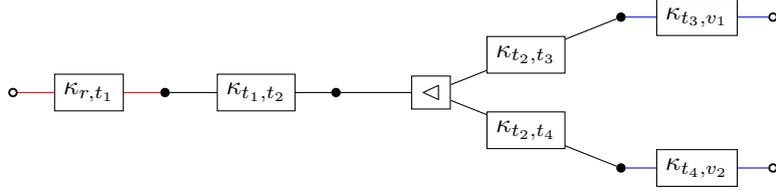


Figure 3: String diagram corresponding to Figure 2. \triangleleft is the duplication kernel defined in (2.5).

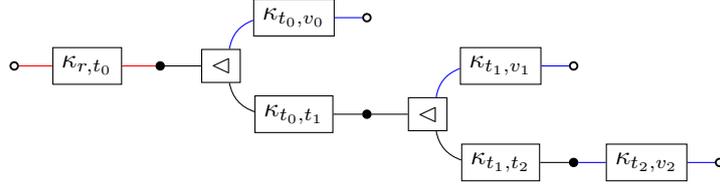


Figure 4: String diagram corresponding to Figure 1 when $n = 2$.

$i - 1$ to i is governed by the Markov kernel $\kappa_i : E_{i-1} \rightarrow E_i$. Then, if at node 0 the process equals the known quantity \hat{x}_0 , the distribution at vertex 1 is $\delta_{\hat{x}_0} \kappa_1$, the pushforward of $\delta_{\hat{x}_0}$ (cf. Equation (2.1)). Pushing forward iteratively, we see that the random quantity at vertex i , denoted by X_i , has distribution

$$\delta_{\hat{x}_0} \kappa_1 \kappa_2 \cdots \kappa_i. \quad (2.6)$$

2.3 Conditioning on a line graph

Suppose that we observe x_{n+1} at vertex $n + 1$. See Figure 5, where we have added the Markov kernels along the branches. The *smoothing distribution* is the distribution of the process on the graph, conditioned on its value x_{n+1} at node $n + 1$. We are interested in generating samples from the smoothing distribution.

We assume that for the final vertex the kernel $\kappa_{n+1}(x, dy)$ has a jointly measurable density $p_{n+1}(x, y)$ with respect to a dominating measure ν :

$$\kappa_{n+1}(x, A) = \int_A p_{n+1}(x, y) \nu(dy).$$

We define the family of functions $(h_i)_{i=1}^n$ as follows.

Definition 1. On a line graph, the *backward filter* is defined by the following recursive scheme, where we recurse from $i = n$ down to $i = 1$:

$$\begin{aligned} h_n(\cdot) &= p_{n+1}(\cdot, x_{n+1}), \\ h_{i-1} &= \kappa_i h_i, \quad i = 1, \dots, n, \end{aligned} \quad (2.7)$$

each h_{i-1} defined as a pullback as in (2.2).



Figure 5: Stochastic process on a line graph with one observation corresponding to the composition of Markov kernels specified in Equation (2.6).

Each $h_i(x)$ is the *likelihood* corresponding to a fixed observation $X_{n+1} = x_{n+1}$, given ‘parameter’ $X_i = x$. The stochastic process that is conditioned on $X_{n+1} = x_{n+1}$ is Markovian then with transition kernels

$$\kappa_i^*(x, dy) = \frac{\kappa_i(x, dy)h_i(y)}{\int_{E_i} \kappa_i(x, dy')h_i(y')}. \quad (2.8)$$

This motivates the following definition.

Definition 2. For a family of strictly positive functions (h_i) , the *h-transform* maps each (κ_i, h_i) to κ_i^* , where κ_i^* is defined in (2.8).

Therefore, if $(h_i)_{i=0}^n$ are known, we can pushforward $\delta_{\hat{x}_0}$, via

$$\delta_{\hat{x}_0} \kappa_1^* \kappa_2^* \cdots \kappa_n^*, \quad (2.9)$$

which gives the marginal distribution of the conditioned process at vertex i , with the joint distribution of (X_0^*, \dots, X_n^*) being

$$P^*(dx_0, \dots, dx_n) = \delta_{\hat{x}_0}(dx_0) \kappa_1^*(x_0, dx_1) \cdots \kappa_n^*(x_{n-1}, dx_n).$$

We remark that, with marginal distributions of X_i given by (2.6) and here denoted by μ_i , for almost all observations (with respect to the marginal distribution of X_{n+1}), the functions $h_i(x)$ will be different from 0 on sets that have positive marginal probability (with respect to μ_i). Therefore κ_i^* is well-defined up sets of zero marginal probability.

We conclude that a realisation of the conditioned process can be obtained as follows: first compute the *backward* filter, to obtain the family $(h_i)_{i=0}^n$, then *forward* simulate from the kernels κ_i^* , where κ_i^* is the *h-transform* of (κ_i, h_i) . Hence, the operation is *bidirectional*. This is spelt out explicitly in the Backward Filtering Forward Sampling algorithm detailed in Algorithm 1.

2.4 Guiding on a line graph

In most cases of practical interest, the backward filter (2.7) cannot be computed in closed form. For that reason, we associate to each kernel κ_i an approximate *guiding* kernel $\tilde{\kappa}_i$ which is chosen such that the backward filter with the kernels $\tilde{\kappa}_i$ is tractable (that is, more easily computed, preferably in closed form):

$$\begin{aligned} g_n(\cdot) &= \tilde{p}_{n+1}(\cdot, x_{n+1}) \\ g_{i-1} &= \tilde{\kappa}_i g_i, \quad i = 1, \dots, n. \end{aligned} \quad (2.10)$$

We will refer to these functions $\{g_i\}_{i=0}^n$ as the *guiding functions*. Using our guiding functions, we define the Markov kernels obtained by the *h-transform* of (κ_i, g_i) :

$$\kappa_i^\circ(x, dy) = \frac{\kappa_i(x, dy)g_i(y)}{\int_{E_i} \kappa_i(x, dy')g_i(y')}. \quad (2.11)$$

If chosen properly (see Remark 2), the sequence of maps $\{g_i\}_{i=0}^n$ ensures that realisations of the process under κ_i° are *guided* towards x_{n+1} at vertex $n+1$. Indeed, these kernels κ_i° define a *guided process* with law P° (2.13) with the property that our observed point x_{n+1} has high probability under P° . By taking quotients of (2.8) and (2.11), we obtain Radon–Nikodym derivatives

$$\frac{\kappa_i^*(x, dy)}{\kappa_i^\circ(x, dy)} = \frac{h_i(y)}{g_i(y)} \frac{\int_{E_i} \kappa_i(x, dy')g_i(y')}{\int_{E_i} \kappa_i(x, dy')h_i(y')}.$$

Proposition 1. The likelihood ratio of paths samples under κ_i^* relative to κ_i° is given by

$$L(x_1, \dots, x_n) = \frac{h_n(x_n)}{g_n(x_n)} \frac{g_0(\hat{x}_0)}{h_0(\hat{x}_0)} \prod_{i=1}^n \frac{\kappa_i g_i(x_{i-1})}{\tilde{\kappa}_i g_i(x_{i-1})}.$$

Algorithm 1 Backward Filtering Forward Sampling

Input: Initial state x_0 ; observation time $n + 1$; transition kernels κ_i ; observation x_{n+1} of X_{n+1} ; conditional ν -density p_{n+1} .

Output: A sample path (x_0, x_1, \dots, x_n) from P^* and likelihood $h_0(x_0)$.

```
1: function BACKWARDFILTER( $(\kappa_1, \dots, \kappa_n), h_n, n$ )
2:   for  $i = n$  down to 1 do
3:      $h_{i-1} \leftarrow \kappa_i h_i$ 
4:   end for
5:    $\mathbf{m} \leftarrow (h_1, \dots, h_n)$ 
6:   return  $h_0, \mathbf{m}$ 
7: end function

8: function FORWARDSAMPLE1( $x, \kappa, h$ )
9:   Let  $\pi(dy) \propto \kappa(x, dy)h(y)$ 
10:  Sample  $y \sim \pi$ 
11:  return  $y$ 
12: end function

13: function FORWARDSAMPLE( $x_0, (\kappa_1, \dots, \kappa_n), \mathbf{m}, n$ )
14:   $(h_1, \dots, h_n) \leftarrow \mathbf{m}$ 
15:  for  $i = 1$  to  $n$  do
16:     $x_i \leftarrow \text{FORWARDSAMPLE1}(x_{i-1}, \kappa_i, h_i)$ 
17:  end for
18:  return  $(x_0, \dots, x_n)$ 
19: end function

20: function BFFS( $x_0, (\kappa_1, \dots, \kappa_n), p_{n+1}, x_{n+1}, n$ )
21:   $h_n(\cdot) \leftarrow p_{n+1}(\cdot, x_{n+1})$ 
22:   $h_0, \mathbf{m} \leftarrow \text{BACKWARDFILTER}((\kappa_1, \dots, \kappa_n), h_n, n)$ 
23:  return  $\text{FORWARDSAMPLE}(x_0, (\kappa_1, \dots, \kappa_n), \mathbf{m}, n), h_0(x_0)$ 
24: end function
```

Proof. The likelihood ratio is

$$\begin{aligned} L(x_1, \dots, x_n) &= \prod_{i=1}^n \frac{h_i(x_i)}{g_i(x_i)} \frac{\int \kappa_i(x_{i-1}, dy) g_i(y)}{\int \kappa_i(x_{i-1}, dy) h_i(y)} \\ &= \frac{h_n(x_n)}{h_0(\hat{x}_0)} \prod_{i=1}^n \frac{\int \kappa_i(x_{i-1}, dy) g_i(y)}{g_i(x_i)} \\ &= \frac{h_n(x_n)}{g_n(x_n)} \frac{g_0(\hat{x}_0)}{h_0(\hat{x}_0)} \prod_{i=1}^n \frac{\kappa_i g_i(x_{i-1})}{\tilde{\kappa}_i g_i(x_{i-1})}, \end{aligned}$$

where the first equality is simply the definitions, the second equality is due to the cancellation of terms due from the recursive relation in (2.7) and the final equality from the recursive relation (2.10). \square

Proposition 1 implies that for bounded functions f ,

$$\mathbb{E} [f(X_1, \dots, X_n) \mid X_{n+1} = x_{n+1}] = \mathbb{E}^\circ [f(X_1, \dots, X_n) L(X_1, \dots, X_n)], \quad (2.12)$$

where \mathbb{E}° denotes expectation under which X has law

$$P^\circ(dx_0, \dots, dx_n) = \delta_{\hat{x}_0}(dx_0) \kappa_1^\circ(x_0, dx_1) \cdots \kappa_n^\circ(x_{n-1}, dx_n). \quad (2.13)$$

Assuming $h_n = \tilde{p}_{n+1}(\cdot, x_{n+1})$ to be tractable and $X_0 = \hat{x}_0$ being fixed and known, the likelihood ratio does not involve h_1, \dots, h_{n-1} which may be hard or expensive to compute. It does however depend on $h_0(\hat{x}_0)$, the *evidence*, which can usually not be evaluated. Fortunately, Markov Chain or Sequential Monte Carlo methods that exploit the relation in (2.12) only require evaluation of L up to a fixed multiplicative constant. For instance, (2.12) implies

$$\mathbb{E} [f(X_1, \dots, X_n) \mid X_{n+1} = x_{n+1}] = \frac{\mathbb{E}^\circ [f(X_1, \dots, X_n) L(X_1, \dots, X_n)]}{\mathbb{E}^\circ [L(X_1, \dots, X_n)]}.$$

The right-hand-side can be approximated by the *self-normalised importance sampling estimator*,

$$T := \frac{\sum_{i=1}^I f(X_{1,i}, \dots, X_{n,i}) L(X_{1,i}, \dots, X_{n,i})}{\sum_{i=1}^I L(X_{1,i}, \dots, X_{n,i})}, \quad (2.14)$$

where $\{(X_{1,i}, \dots, X_{n,i})\}_{i=1}^I$ are independent samples under P° . We see that T does not depend on $h_0(\hat{x}_0)$.

An algorithmic description of the full Backward Filtering Forward Guiding is given in Algorithm 2. In the case when $\tilde{\kappa} = \kappa$, we recover BFFS as in Algorithm 1.

Remark 2. We need to choose the guiding functions $\{g_i\}$ such that P^* is absolutely continuous with respect to P° . A specific choice can be made for example by minimising the (reverse) Kullback–Leibler divergence between P^* and P° ; other discrepancy measures between the two measures are possible. Alternatively, if one uses BFFG within a sequential Monte Carlo algorithm, a practical criterion consists of choosing the guiding functions such that the effective sample size of particles is maximised.

Remark 3. For ease of exposition, we have introduced the BFFG algorithm on a line graph with a single observation. We refer to Theorem 2.2 in Van der Meulen and Schauer [2022] for a more general statement on a directed acyclic graph.

3 Definitions from category theory

The main contribution of this work is to connect the BFFG algorithm described above with well-established concepts from category theory. Our main contribution will be showing that the BFFG algorithm *preserves sequential composition* and furthermore *parallel composition* in the case of BFFS. To formalise this, we will make use of well-established categorical concepts such as *monoidal categories* and *lax monoidal functors*. For readers who are less familiar with these concepts, we have included full definitions in Appendix A.

Algorithm 2 Backward Filtering Forward Guiding (BFFG)

Input: Initial state x_0 ; observation time $n + 1$; transition kernels (κ_i) , $(\tilde{\kappa}_i)$; observation x_{n+1} of X_{n+1} ; conditional ν -densities p_{n+1} and \tilde{p}_{n+1} .

Output: A sample path (x_0, x_1, \dots, x_n) from P° and a weight $\varpi = h_0(x_0) \cdot L(x_1, \dots, x_n)$.

```
1: function FORWARDGUIDING1( $(x, \varpi), \kappa, \tilde{\kappa}, g$ )
2:   Let  $\pi(dy) = \kappa(x, dy)g(y)/(\kappa g)(x)$ 
3:   Sample  $y \sim \pi$ 
4:    $w \leftarrow \frac{(\kappa g)(x)}{(\tilde{\kappa} g)(x)}$ 
5:   return  $y, w \cdot \varpi$ 
6: end function

7: function FORWARDGUIDING( $(x_0, \varpi), (\kappa_1, \dots, \kappa_n), (\tilde{\kappa}_1, \dots, \tilde{\kappa}_n), \mathbf{m}, n$ )
8:    $(g_1, \dots, g_n) \leftarrow \mathbf{m}$ 
9:   for  $i = 1$  to  $n$  do
10:     $x_i, \varpi \leftarrow$  FORWARDGUIDING1( $(x_{i-1}, \varpi), \kappa_i, \tilde{\kappa}_i, g_i$ )
11:   end for
12:   return  $(x_0, \dots, x_n), \varpi$ 
13: end function

14: function BFFG( $x_0, (\kappa_1, \dots, \kappa_n), (\tilde{\kappa}_1, \dots, \tilde{\kappa}_n), p_{n+1}, \tilde{p}_{n+1}, x_{n+1}, n$ )
15:    $h_n(\cdot) = p_{n+1}(\cdot, x_{n+1})$ 
16:    $g_n(\cdot) = \tilde{p}_{n+1}(\cdot, x_{n+1})$ 
17:    $g_0, \mathbf{m} \leftarrow$  BACKWARDFILTER( $(\tilde{\kappa}_1, \dots, \tilde{\kappa}_n), g_n, n$ )
18:    $(x_0, \dots, x_n), \varpi \leftarrow$  FORWARDGUIDING( $(x_0, g_0(x_0)), (\kappa_1, \dots, \kappa_n), (\tilde{\kappa}_1, \dots, \tilde{\kappa}_n), \mathbf{m}, n$ )
19:   return  $(x_0, \dots, x_n), \varpi \cdot h_n(x_n)/g_n(x_n)$ 
20: end function
```

3.1 The category Stoch

It is a key contribution of categorical probability, Fritz [2020], Cho and Jacobs [2019], that the Chapman–Kolmogorov equations (2.3), actually constitute composition $\kappa_2 \circ \kappa_1$ in a particular category, Stoch, of Markov kernels. Formally, we have the following well-known fact; see, for instance [Fritz, 2020, Section 4]:

Proposition 4. The category Stoch, with objects measurable spaces, S, S', \dots , and Markov kernels $\kappa: S \rightarrow S'$ as morphisms is a symmetric monoidal category: sequential composition is as defined in (2.3). The identity id for this composition is the identity function considered as a Markov kernel, $\text{id}(x, dy) = \delta_x(dy)$. The monoidal product is as in (2.4) with monoidal unit $1_{\text{Stoch}} = (\{*\}, (\emptyset, \{*\}))$.

3.2 Optics

Optics were introduced in Riley [2018] for modelling systems with bidirectional data flow. Here, we present the formulation as given in Gavranović [2022]. Note that an optic models a transformation in which a first (forward) pass is followed by a second (backward) pass.

Definition 3. Given a symmetric monoidal base category $(\mathcal{C}, \otimes, 1, \alpha, \lambda, \rho)$, an **optic** from a pair of objects from \mathcal{C} , (A, A') to a pair (B, B') is an equivalence class of triplets $(M, \text{fm}, \text{sm})$ where

- M is an object of \mathcal{C} , the type of the *internal state*;
- $\text{fm}: A \rightarrow M \otimes B$, a morphism referred to as the *first map*;
- $\text{sm}: M \otimes B' \rightarrow A'$, a morphism referred to as the *second map*;

and the equivalence class is given by the finest equivalence relation containing the following relation: for two such triplets (M, f, f') and (N, g, g') , we have $(M, f, f') \sim (N, g, g')$ if there exists a *residual morphism* $r: M \rightarrow N$ such that $(r \otimes B) \circ f = g$ and $g' \circ (r \otimes B') = f'$:

$$\begin{array}{ccc} A & \xrightarrow{f} & M \otimes B \\ & \searrow g & \downarrow r \otimes B \\ & & N \otimes B \end{array} \quad \begin{array}{ccc} M \otimes B' & \xrightarrow{f'} & A' \\ r \otimes B' \downarrow & \nearrow g' & \\ N \otimes B' & & \end{array} .$$

An optic from (A, A') to (B, B') can be visualised as follows:

$$\begin{array}{ccccc} A & \longrightarrow & \boxed{\text{fm}} & \longrightarrow & B \\ & & \downarrow M & & \\ A' & \longleftarrow & \boxed{\text{sm}} & \longleftarrow & B' \end{array} \quad (3.1)$$

In the first pass, fm produces a *message* of type M , the internal state, which is consumed in the second pass by sm.

Remark 5. In the literature, the first pass is typically referred to as the *forward* pass, and the second pass as the *backward* pass, reflecting the flow of the information, e.g. Gavranović [2022]. We have opted to use the (new) terminology of first/second instead of forwards/backwards in order to avoid confusion: the first (i.e. forward) pass of the BFFG algorithm is actually the *backward* filter, and the second (i.e. backward) pass of the BFFG algorithm is the *forward* guiding step.

Definition 4 (Composition of optics). Optics can be composed as follows [Gavranović, 2022, Definition 3]: consider two optics

$$\begin{bmatrix} A \\ A' \end{bmatrix} \xrightarrow{(M_1, \text{fm}_1, \text{sm}_1)} \begin{bmatrix} B \\ B' \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} B \\ B' \end{bmatrix} \xrightarrow{(M_2, \text{fm}_2, \text{sm}_2)} \begin{bmatrix} C \\ C' \end{bmatrix},$$

then the composite $(M, \text{fm}, \text{sm}): (A, A') \rightarrow (C, C')$ is defined by

$$M := M_1 \otimes M_2 \quad (3.2)$$

$$\text{fm} := (\text{id}_{M_1} \otimes \text{fm}_2) \circ \text{fm}_1: A \rightarrow M_1 \otimes (M_2 \otimes C) \quad (3.3)$$

$$\text{sm} := \text{sm}_1 \circ (\text{id}_{M_1} \otimes \text{sm}_2): M_1 \otimes (M_2 \otimes C') \rightarrow A', \quad (3.4)$$

noting we have further isomorphisms $M_1 \otimes (M_2 \otimes C) \cong (M_1 \otimes M_2) \otimes C$ and $M_1 \otimes (M_2 \otimes C') \cong (M_1 \otimes M_2) \otimes C'$. This composition is represented diagrammatically below. The blue rectangles indicate that its constituent components should be viewed together as a single unit; compare with (3.1).

$$\begin{array}{ccccc} A & \longrightarrow & \boxed{\begin{array}{ccc} \text{fm}_1 & \xrightarrow{B} & \text{fm}_2 \end{array}} & \longrightarrow & C \\ & & \downarrow M_1 & & \downarrow M_2 \\ A' & \longleftarrow & \boxed{\begin{array}{ccc} \text{sm}_1 & \xleftarrow{B'} & \text{sm}_2 \end{array}} & \longleftarrow & C' \end{array}$$

Definition 5 (Tensor product of optics). Consider two optics

$$\begin{bmatrix} A_1 \\ A'_1 \end{bmatrix} \xrightarrow{(M_1, \text{fm}_1, \text{sm}_1)} \begin{bmatrix} B_1 \\ B'_1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} A_2 \\ A'_2 \end{bmatrix} \xrightarrow{(M_2, \text{fm}_2, \text{sm}_2)} \begin{bmatrix} B_2 \\ B'_2 \end{bmatrix},$$

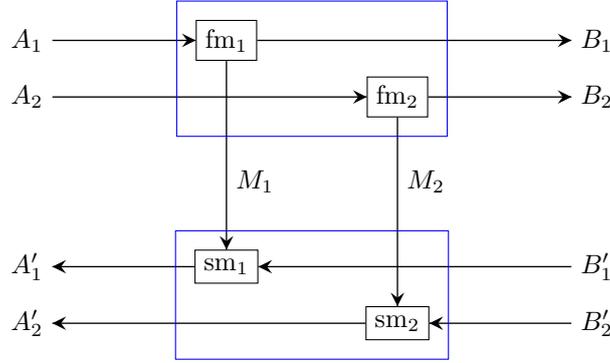
then the tensor product $(M, \text{fm}, \text{sm})$, mapping

$$\begin{bmatrix} A_1 \otimes A_2 \\ A'_1 \otimes A'_2 \end{bmatrix} \xrightarrow{(M, \text{fm}, \text{sm})} \begin{bmatrix} B_1 \otimes B_2 \\ B'_1 \otimes B'_2 \end{bmatrix}$$

is defined by

$$\begin{aligned} M &:= M_1 \otimes M_2 \\ \text{fm} &:= \text{fm}_1 \otimes \text{fm}_2: A_1 \otimes A_2 \rightarrow (M_1 \otimes B_1) \otimes (M_2 \otimes B_2) \\ \text{sm} &:= \text{sm}_1 \otimes \text{sm}_2: (M_1 \otimes B'_1) \otimes (M_2 \otimes B'_2) \rightarrow A'_1 \otimes A'_2, \end{aligned}$$

further noting that since the base category is symmetric monoidal we have isomorphisms $(M_1 \otimes B_1) \otimes (M_2 \otimes B_2) \cong (M_1 \otimes M_2) \otimes (B_1 \otimes B_2)$ and $(M_1 \otimes B'_1) \otimes (M_2 \otimes B'_2) \cong (M_1 \otimes M_2) \otimes (B'_1 \otimes B'_2)$. This tensor product can be visualised by placing the corresponding diagrams (3.1) adjacent and overlapping one another; again, blue rectangles indicate those sections should be viewed as a single block:



Definition 6 (Identity optic). Given (A, A') , the identity optic $\text{id}_{A, A'}^{\text{O}}: (A, A') \rightarrow (A, A')$ is $(1, \lambda_A^{-1}, \lambda_{A'})$: we take as internal state 1 , the monoidal unit, then the first map is $\lambda_A^{-1}: A \rightarrow 1 \otimes A$ and the second map is $\lambda_{A'}: 1 \otimes A' \rightarrow A'$, the left unitors.

Definition 7 (Category of optics). The category $\text{Optic}(\mathcal{C})$ has as objects pairs of objects from \mathcal{C} , (A, A') , (B, B') , \dots , and as morphisms optics $(M, \text{fm}, \text{sm})$ from (A, A') to (B, B') as defined in Definition 3. Sequential composition, the monoidal product and identities are as defined above, and this gives rise to a symmetric monoidal category.

In the case of $\text{Optic}(\text{Set})$, the monoidal unit is simply $\{*\} \otimes \{*\}$.

See [Riley, 2018, Theorem 2.0.12] for a proof that $\text{Optic}(\mathcal{C})$ is symmetric monoidal.

Remark 6. For our subsequent compositional results for BFFG, we will be working with $\text{Optic}(\text{Set})$. In this case, since Set is Cartesian, we have the isomorphism $\text{Optic}(\text{Set}) \cong \text{Lens}(\text{Set})$ [Gavranović, 2022, Proposition 1], so we could alternatively have worked with lenses.

4 A categorical approach to Backward Filtering Forward Guiding

In Section 2, we introduced the BFFG algorithm on a line graph. It consists of two steps:

- first, in the *backward pass*, we iteratively compute $g_{i-1} = \tilde{\kappa}g_i$;
- secondly, in the *forward pass* we iteratively apply the h -transform from Definition 2 to (κ_i, g_i) .

In this section, we will formalise both steps by defining formally first and second maps, using the framework of Optics as defined in Section 3.2. The first pass will correspond to the backward pass of BFFG, and the second pass will correspond to the forward pass of BFFG.

4.1 First and second maps

In the upcoming definitions we assume we are given Markov kernels $\kappa: S \rightarrow S'$ and $\tilde{\kappa}: S \rightarrow S'$, where $S = (E, \mathfrak{B})$ and $S' = (E', \mathfrak{B}')$ are measurable spaces. Recall that a measure μ is *s-finite* if there exists a countable family of finite measures (μ_n) with $\mu = \sum_n \mu_n$.

For a measurable space $S = (E, \mathfrak{B})$, let $\mathbf{M}_+(S)$ denote the set of strictly positive measurable functions on S . Let $\mathcal{M}_s(S)$ denote the set of s-finite measures on S . The class of s-finite measures is convenient to work with, since it is closed under *pushforwards* of measurable maps and Markov kernels [Staton, 2017, Proposition 7].

Definition 8. For a Markov kernel $\tilde{\kappa}: S \rightarrow S'$, define $\mathcal{B}_{\tilde{\kappa}}: \mathbf{M}_+(S') \rightarrow \mathbf{M}_+(S') \times \mathbf{M}_+(S)$ by

$$\mathcal{B}_{\tilde{\kappa}}(g) = (g, \tilde{\kappa}g). \quad (4.1)$$

Intuitively, we should imagine that the kernel $\tilde{\kappa}$ is an approximation to the kernel κ , and then g represents an approximation to the functions h defined through (2.7).

This map returns both the pullback $\tilde{\kappa}g$ and an appropriate *message* g for the map $\mathcal{F}_{\tilde{\kappa}, \kappa}$ specified in the following definition. This corresponds to the backwards map in BFFG, and will be the first map, fm, in terms of the optic (Definition 3).

Definition 9. For Markov kernels $\kappa, \tilde{\kappa}: S \rightarrow S'$, define $\mathcal{F}_{\tilde{\kappa}, \kappa}: \mathbf{M}_+(S') \times \mathcal{M}_s(S) \rightarrow \mathcal{M}_s(S')$ by

$$\mathcal{F}_{\tilde{\kappa}, \kappa}(g, \mu) = \nu, \quad \text{with} \quad \nu(dy) := \int_S \frac{g(y)}{\tilde{\kappa}g(x)} \mu(dx) \kappa(x, dy). \quad (4.2)$$

We note that $\mathcal{F}_{\tilde{\kappa}, \kappa}$ is well-defined by considering the following decomposition: given $\mu = \sum_n \mu_n$ for finite measures (μ_n) , consider for each $s \in \mathbb{N}$, $B_s := \{(x, y): s \leq m(x, y) < s + 1\} \subset S \otimes S'$ with $m(x, y) := \frac{g(y)}{\kappa g(x)}$; this is well-defined by strict positivity of g . Then, $\nu_{n,s}(dy) := \int m(x, y) \mu_n(dx) \kappa(x, dy) 1_{B_s}(x, y)$ is a finite measure, with $\sum_{n,s} \nu_{n,s} = \nu$.

This corresponds to the forwards pass in BFFG, and in terms of the optic will be the second map, sm, Definition 3.

The pairing of the first and second maps can be conveniently summarised by the following diagram, which is analogous to (3.1):

$$\begin{array}{ccccc} \mathbf{B}_1 & \longrightarrow & \boxed{\mathcal{B}_{\tilde{\kappa}}} & \longrightarrow & \mathbf{B}_0 \\ & & \downarrow g & & \\ \mathbf{M}_1 & \longleftarrow & \boxed{\mathcal{F}_{\tilde{\kappa}, \kappa}} & \longleftarrow & \mathbf{M}_0 \end{array}$$

Note that in the case $\tilde{\kappa} = \kappa$, if μ is a probability measure, then $\mathcal{F}_{\tilde{\kappa}, \kappa}(g, \mu)$ is a probability measure as well. In general, even if μ is a probability measure, the output of $\mathcal{F}_{\tilde{\kappa}, \kappa}$ will typically *not* be a probability measure. In that case, we can reinterpret $\mathcal{F}_{\tilde{\kappa}, \kappa}$ as producing a *weighted* probability measure: if $\varpi \geq 0$ and μ is a probability measure, then

$$\mathcal{F}_{\tilde{\kappa}, \kappa}(g, \varpi \cdot \mu)(dy) = (\varpi w_{\tilde{\kappa}, \kappa}(g, \mu)) \cdot \nu(dy),$$

where the *weight* $w_{\tilde{\kappa}, \kappa}(g, \mu)$ and probability measure ν are defined by

$$\begin{aligned} w_{\tilde{\kappa}, \kappa}(g, \mu) &= \int_S \int_{S'} \frac{g(y)}{\tilde{\kappa}g(x)} \kappa(x, dy) \mu(dx) = \int_S \frac{(\kappa g)(x)}{(\tilde{\kappa}g)(x)} \mu(dx) \\ \nu(dy) &= w_{\tilde{\kappa}, \kappa}^{-1}(g, \mu) \int_S \frac{g(y)}{(\tilde{\kappa}g)(x)} \mu(dx) \kappa(x, dy), \end{aligned} \quad (4.3)$$

provided we have $0 < w_{\tilde{\kappa}, \kappa}(m, \mu) < \infty$. Hence, by using an approximate $\tilde{\kappa}$ in $\mathcal{B}_{\tilde{\kappa}}$, we “pick up” a weight $w_{\tilde{\kappa}, \kappa}(g, \mu)$.

Definition 10. Given kernels $\tilde{\kappa}, \kappa: S \rightarrow S'$, our triple $(\mathbf{M}_+(S'), \mathcal{B}_{\tilde{\kappa}}, \mathcal{F}_{\tilde{\kappa}, \kappa})$ defines an optic $(\mathbf{M}_+(S'), \mathcal{M}_s(S')) \rightarrow (\mathbf{M}_+(S), \mathcal{M}_s(S))$ in the category $\text{Optic}(\text{Set})$. This optic will be denoted by $\mathcal{O}(\kappa, \tilde{\kappa})$.

Remark 7. The BFFG algorithm (as in Algorithm 2) is related to our above construction as follows:

- Given $g_{n+1} = \tilde{p}_{n+1}(\cdot, x_{n+1})$ analogous to (2.7), the backward filtering step of Algorithm 2, Line 17 is equivalent to the composition of first maps $\mathcal{B}_{\tilde{\kappa}_1}, \dots, \mathcal{B}_{\tilde{\kappa}_n}(g_{n+1})$, using the composition rule (3.3). This produces a sequence of guiding functions (g_0, \dots, g_n) needed for the second pass.
- The forward guiding step of Algorithm 2, Line 18 is equivalent to sampling from the compositions of second maps $\mathcal{F}_{\tilde{\kappa}_n, \kappa_n}, \dots, \mathcal{F}_{\tilde{\kappa}_1, \kappa_1}(\delta_{\tilde{x}_0})$ using the composition rule (3.4), given the output (g_0, \dots, g_n) of the first maps.

Thus the overall BFFG algorithm is precisely given by composing the optics $(\mathbf{M}_+(S_i), \mathcal{B}_{\tilde{\kappa}_i}, \mathcal{F}_{\tilde{\kappa}_i, \kappa_i})_{i=1}^n$ as given in Definition 10.

Definition 11. Since we will be working with pairs of kernels $\kappa, \tilde{\kappa}$, we define the category Stoch_2 to be the category whose objects are measurable spaces, as in Stoch , but whose morphisms are now *ordered pairs* of kernels $(\kappa, \tilde{\kappa}): S \rightarrow S'$ with the same source and target. Composition is then performed component-wise: $(\kappa, \tilde{\kappa}) \circ (\kappa', \tilde{\kappa}') = (\kappa \circ \kappa', \tilde{\kappa} \circ \tilde{\kappa}')$, and similarly the monoidal product is also defined pairwise on each component. The required associativity and identities are directly inherited from Stoch .

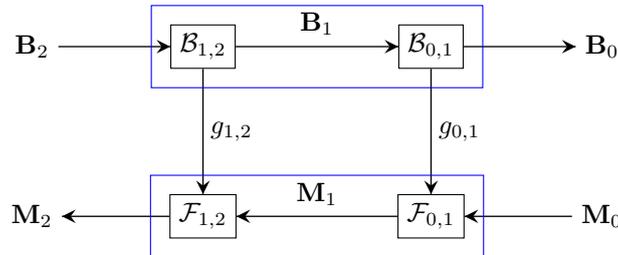
4.2 Functoriality of \mathcal{O} : equivalence of sequential composition

It turns out that the BFFG algorithm preserves sequential composition from kernels to optics. Formally, we have the following:

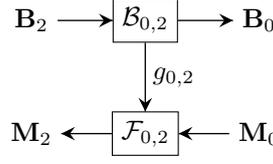
Theorem 8. The BFFG algorithm defines a contravariant functor $\mathcal{O}: (\text{Stoch}_2)^{\text{Op}} \rightarrow \text{Optic}(\text{Set})$:

- on objects, given a measurable space S , we have $\mathcal{O}(S) = (\mathbf{M}_+(S), \mathcal{M}_s(S))$;
- on arrows, given $(\kappa, \tilde{\kappa}): S \rightarrow S'$, we have $\mathcal{O}(\kappa, \tilde{\kappa}): \mathcal{O}(S') \rightarrow \mathcal{O}(S)$ is as defined in Definition 10.

To spell this out, consider the composition of two Markov kernels, say $\kappa_{0,1}$ and $\kappa_{1,2}$, and corresponding kernels $\tilde{\kappa}_{0,1}$ and $\tilde{\kappa}_{1,2}$. We introduce the short-hand notation $\mathcal{B}_{i,i+1} = \mathcal{B}_{\tilde{\kappa}_{i,i+1}}$ and $\mathcal{F}_{i,i+1} = \mathcal{F}_{\tilde{\kappa}_{i,i+1}, \kappa_{i,i+1}}$ for $i \in \{0,1\}$. The optics $\mathcal{O}(\kappa_{0,1}, \tilde{\kappa}_{0,1})$ and $\mathcal{O}(\kappa_{1,2}, \tilde{\kappa}_{1,2})$ induced by $(\tilde{\kappa}_{0,1}, \kappa_{0,1})$ and $(\tilde{\kappa}_{1,2}, \kappa_{1,2})$ respectively can be composed as follows:



However, since Markov kernels can be composed – using the Chapman-Kolmogorov equation (2.3) – there is an alternative way to compose. Defining the kernels $\mathcal{B}_{0,2} = \mathcal{B}_{\tilde{\kappa}_{0,2}}$, $\mathcal{F}_{0,2} = \mathcal{F}_{\tilde{\kappa}_{0,2}, \kappa_{0,2}}$ with composed kernels $\kappa_{0,2} = \kappa_{0,1}\kappa_{1,2}$ and $\tilde{\kappa}_{0,2} = \tilde{\kappa}_{0,1}\tilde{\kappa}_{1,2}$, we have



Hence, there are two ways to compose:

1. **construction 1:** compose Markov kernels to get $\kappa_{0,2} = \kappa_{0,1}\kappa_{1,2}$ (similarly $\tilde{\kappa}_{0,2} = \tilde{\kappa}_{0,1}\tilde{\kappa}_{1,2}$), then form the optic;
2. **construction 2:** compose optics $(\mathbf{M}_+(S'), \mathcal{B}_{0,1}, \mathcal{F}_{0,1})$ and $(\mathbf{M}_+(S''), \mathcal{B}_{1,2}, \mathcal{F}_{1,2})$.

The content of Theorem 8 is that these two constructions give rise to the same optic. The proof is broken into two steps.

Lemma 9. Both optics are in the same equivalence class. That is

$$\mathcal{O}(\kappa_{0,2}, \tilde{\kappa}_{0,2}) \sim \mathcal{O}(\kappa_{1,2}, \tilde{\kappa}_{1,2}) \circ \mathcal{O}(\kappa_{0,1}, \tilde{\kappa}_{0,1}),$$

the right-hand-side denoting composition of optics.

Proof. For the construction 1, we see that it has internal state $M := \mathbf{M}_+(S'')$, with

$$\mathcal{B}_{0,2}(g) = (g, \tilde{\kappa}_{0,2}g), \quad g \in \mathbf{M}_+(S''),$$

and for $g \in \mathbf{M}_+(S'')$ and $\mu \in \mathcal{M}_s(S)$,

$$\begin{aligned}
\mathcal{F}_{0,2}(g, \mu) &= \nu, \\
\nu(dy) &= \int_S \frac{g(y)}{\tilde{\kappa}_{0,2}g(x)} \mu(dx) \kappa_{0,2}(x, dy).
\end{aligned}$$

On the other hand, with construction 2 by composing optics we find we have inner state $N := \mathbf{M}_+(S'') \otimes \mathbf{M}_+(S')$ and maps

$$\mathcal{B}'_{0,2}: g \mapsto (g, \tilde{\kappa}_{1,2}g, \tilde{\kappa}_{0,2}g), \quad g \in \mathbf{M}_+(S''),$$

and for $(g, g', \mu) \in \mathbf{M}_+(S'') \otimes \mathbf{M}_+(S') \otimes \mathcal{M}_s(S)$,

$$\begin{aligned}
\mathcal{F}'_{0,2}: (g, g', \mu) &\mapsto \nu', \\
\nu'(dy) &= \int_S \frac{\mu(dx)}{\tilde{\kappa}_{0,1}g'(x)} \int_{S'} \kappa_{0,1}(x, dy') \cdot \frac{g'(y')}{\tilde{\kappa}_{1,2}g(y')} \cdot \kappa_{1,2}(y', dy) g(y). \tag{4.4}
\end{aligned}$$

To show that these are in the same equivalence class, we define the residual morphism $r: M \rightarrow N$ via

$$r: \mathbf{M}_+(S'') \ni g \mapsto (g, \tilde{\kappa}_{1,2}g) \in \mathbf{M}_+(S'') \otimes \mathbf{M}_+(S').$$

With this choice of r , it is easy to see that the required commutation relations on r in Definition 3 do hold; for instance in the definition of ν' in (4.4), given $g' = \tilde{\kappa}_{1,2}g$, the inner fraction simplifies to 1 and we see $\nu' = \nu$ in this case. \square

In order to complete the proof of Theorem 8, we also need to establish preservation of identities.

Lemma 10. For any measure space S , writing id_S for the identity kernel $x \mapsto \delta_x$ in Stoch, we have that $\mathcal{O}(\text{id}_S, \text{id}_S)$ is the identity optic $\text{id}_{\mathcal{O}(S)}$ in $\text{Optic}(\text{Set})$.

Proof. For the first pass, we have $\mathcal{B}_{\text{id}_S}(g) = (g, g)$. For the second pass, we have $\mathcal{F}_{\text{id}_S, \text{id}_S}(g, \mu) = \nu$ with $\nu(dx) = \mu(dx)$.

By trivially discarding the message g , that is, $r: \mathbf{M}_+(S) \rightarrow 1 = \{*\}$ maps $g \mapsto *$, we see this corresponds to the identity optic on $(\mathbf{M}_+(S), \mathcal{M}_s(S))$. \square

4.3 Equivalence of parallel composition

In the previous section we established that BFFG is functorial, namely it preserves sequential composition. We now show that BFFS (when $\tilde{\kappa} = \kappa$) is a *lax monoidal functor*: it preserves parallel composition. Intuitively, lax monoidality of BFFS is related to the fact that conditional independence is preserved under conditioning on the leaves. We would not expect more than lax monoidality of BFFS (for instance, bilaxity) due to the phenomenon of confounding: conditioning on joint observations introduces dependence between formerly independent states. The failure of lax monoidality of BFFG is related to the additional need to track global weights; see Remark 17.

Recall (2.4) defining a tensor product of Markov kernels. Intuitively, there are two constructions involving the ‘product’ of two Markov kernels κ_1 and κ_2

1. **construction 1:** take the product of Markov kernels to get $\kappa_{1\otimes 2} = \kappa_1 \otimes \kappa_2$ (similarly for $\tilde{\kappa}_{1\otimes 2} = \tilde{\kappa}_1 \otimes \tilde{\kappa}_2$), then form the corresponding optic;
2. **construction 2:** take the tensor product of the optics $(M_1, \mathcal{B}_1, \mathcal{F}_1)$ and $(M_2, \mathcal{B}_2, \mathcal{F}_2)$.

The goal of this section is to prove our following main result. The crucial restriction here is that we have only a single kernel $\kappa = \tilde{\kappa}$, without an additional guiding kernel: in this setting, the category Stoch_2 can be identified with Stoch .

Theorem 11. In the setting where $\tilde{\kappa} = \kappa$, the BFFG algorithm defines a lax monoidal functor $\mathcal{O}: (\text{Stoch})^{\text{Op}} \rightarrow \text{Optic}(\text{Set})$.

Proof. Since we have already established in Theorem 8 that $\mathcal{O}: (\text{Stoch}_2)^{\text{Op}} \rightarrow \text{Optic}(\text{Set})$ defines a functor, what remains to establish lax monoidal functoriality is defining an appropriate morphism ϵ and natural transformation μ , which together satisfy associativity and unitality.

Recall that the monoidal unit in Stoch is $1_{\text{Stoch}} = (\{*\}, (\emptyset, \{*\}))$ and in $\text{Optic}(\text{Set})$ we have $1_{\text{Optic}(\text{Set})} = \{*\} \otimes \{*\}$.

For each pair of objects S, T in Stoch , define the optic

$$\mu_{S,T}: \mathcal{O}(S) \otimes \mathcal{O}(T) \rightarrow \mathcal{O}(S \otimes T)$$

as follows: we take trivial internal state, and the following first and second maps:

$$\begin{aligned} \text{fm}: \mathbf{M}_+(S) \otimes \mathbf{M}_+(T) &\rightarrow \mathbf{M}_+(S \otimes T), & (g(s), h(t)) &\mapsto g(s) \cdot h(t) \\ \text{sm}: \mathcal{M}_s(S \otimes T) &\rightarrow \mathcal{M}_s(S) \otimes \mathcal{M}_s(T), & \bar{\mu}(ds, dt) &\mapsto (\bar{\mu}(ds, T), \bar{\mu}(S, dt)), \end{aligned}$$

noting that the marginal measures of an s -finite measure remain s -finite.

In other words, a pair of functions on the individual spaces is mapped to the pointwise product function on the joint space, and a measure on the joint space is mapped to its two marginals on the individual spaces.

In order to prove naturality of μ , we need to establish the following naturality square: for kernels $\kappa_1, \tilde{\kappa}_1: S \rightarrow S'$ and $\kappa_2, \tilde{\kappa}_2: T \rightarrow T'$,

$$\begin{array}{ccc} \mathcal{O}(S') \otimes \mathcal{O}(T') & \xrightarrow{\mathcal{O}(\kappa_1; \tilde{\kappa}_1) \otimes \mathcal{O}(\kappa_2; \tilde{\kappa}_2)} & \mathcal{O}(S) \otimes \mathcal{O}(T) \\ \downarrow \mu_{S', T'} & & \downarrow \mu_{S, T} \\ \mathcal{O}(S' \otimes T') & \xrightarrow{\mathcal{O}(\kappa_1 \otimes \kappa_2; \tilde{\kappa}_1 \otimes \tilde{\kappa}_2)} & \mathcal{O}(S \otimes T) \end{array} \quad . \quad (4.5)$$

In Lemma 12 below, we demonstrate that this naturality square does hold in the assumed case when $\tilde{\kappa}_1 = \kappa_1$ and $\tilde{\kappa}_2 = \kappa_2$.

Thus in order to complete the proof of Theorem 11, we need to verify the associativity and unitality conditions of Definition 15.

To this end, define the morphism $\epsilon: 1_{\text{Optic}(\text{Set})} \rightarrow \mathcal{O}(1_{\text{Stoch}}) = \mathbf{M}_+(\{*\}) \otimes \mathcal{M}_s(\{*\})$ to be the optic with trivial internal state and the following first and second maps:

$$\begin{aligned} \text{fm}: \{*\} &\rightarrow \mathbf{M}_+(\{*\}), & * &\mapsto 1 \\ \text{sm}: \mathcal{M}_s(\{*\}) &\rightarrow \{*\}, & \mu &\mapsto *, \end{aligned}$$

that is, in the first pass $*$ is mapped to the constant function 1 and in the second pass the measure is simply deleted.

In Lemma 14, we prove associativity of μ . In Lemma 15, we prove that the unitality conditions on ϵ, μ hold.

Thus the proof of Theorem 11 is complete. \square

Lemma 12. In the case when $\tilde{\kappa}_1 = \kappa_1$ and $\tilde{\kappa}_2 = \kappa_2$, our family of morphisms $\mu_{S,T}$ defines a natural transformation $\mathcal{O}(\cdot) \otimes \mathcal{O}(\cdot) \rightarrow \mathcal{O}(\cdot \otimes \cdot)$.

Proof. Let's compute the two sides of the naturality square (4.5):

- (\downarrow, \rightarrow): here we first map using the component $\mu_{S',T'}$, and then use the BFFG functor on a pair of product kernels. This composite optic has the following components:

$$\begin{aligned} &\text{internal state: } N = \mathbf{M}_+(S' \otimes T'); \\ \text{fm}: \mathbf{M}_+(S') \otimes \mathbf{M}_+(T') &\rightarrow N \otimes \mathbf{M}_+(S \otimes T), \\ &\quad (g_1, g_2) \mapsto (g_1 \cdot g_2, \tilde{\kappa}_1 g_1 \cdot \tilde{\kappa}_2 g_2); \\ \text{sm}: N \otimes \mathcal{M}_s(S \otimes T) &\rightarrow \mathcal{M}_s(S') \otimes \mathcal{M}_s(T'), \\ &\quad (\bar{g}(s', t'), \bar{\mu}(ds, dt)) \mapsto (\nu(ds', T'), \nu(S', dt')), \end{aligned} \tag{4.6}$$

where $\nu(ds', dt') = \int_{S \otimes T} \frac{\bar{g}(s', t')}{[(\tilde{\kappa}_1 \otimes \tilde{\kappa}_2) \bar{g}](s, t)} \bar{\mu}(ds, dt) \kappa_1(s, ds') \kappa_2(t, dt')$.

The distributions (4.6) can be naturally interpreted as the marginal distributions of the output of the guiding pass (the second map).

- (\rightarrow, \downarrow): here we first form the product of two individual optics formed from BFFG, and then use the component $\mu_{S,T}$. We have the following components:

$$\begin{aligned} &\text{internal state: } M = \mathbf{M}_+(S') \otimes \mathbf{M}_+(T'); \\ \text{fm}: \mathbf{M}_+(S') \otimes \mathbf{M}_+(T') &\rightarrow M \otimes \mathbf{M}_+(S \otimes T), \\ &\quad (g_1, g_2) \mapsto (g_1, g_2, \tilde{\kappa}_1 g_1 \cdot \tilde{\kappa}_2 g_2); \\ \text{sm}: M \otimes \mathcal{M}_s(S \otimes T) &\rightarrow \mathcal{M}_s(S') \otimes \mathcal{M}_s(T'), \\ &\quad (g_1, g_2, \bar{\mu}(ds, dt)) \mapsto \left(\int_S \frac{g_1(s')}{\tilde{\kappa}_1 g_1(s)} \bar{\mu}(ds, T) \kappa_1(s, ds'), \int_T \frac{g_2(t')}{\tilde{\kappa}_2 g_2(t)} \bar{\mu}(S, dt) \kappa_2(t, dt') \right). \end{aligned} \tag{4.7}$$

In order to argue that these two optics are equal, we need to present a residual morphism $r: M \rightarrow N$ between the internal states, such that the two triangles from Definition 3 commute. We choose

$$\begin{aligned} r: \mathbf{M}_+(S') \otimes \mathbf{M}_+(T') &\rightarrow \mathbf{M}_+(S' \otimes T'), \\ &\quad (g_1(s'), g_2(t')) \mapsto g_1(s') \cdot g_2(t'). \end{aligned}$$

With this definition, the left-hand triangle of Definition 3 is immediately seen to hold.

For the right-hand triangle, it is equivalent to requiring that the following is equal to the final expression of (4.7) for any $g_1, g_2, \bar{\mu}$:

$$\left(\int_{S \otimes T} \frac{g_1(s')}{\tilde{\kappa}_1 g_1(s)} \bar{\mu}(ds, dt) \kappa_1(s, ds') w_2(t), \int_{S \otimes T} \frac{g_2(t')}{\tilde{\kappa}_2 g_2(t)} \bar{\mu}(ds, dt) \kappa_2(t, dt') w_1(s) \right), \tag{4.8}$$

where

$$w_i(t) := \frac{\kappa_i g_i(t)}{\tilde{\kappa}_i g_i(t)}, \quad i = 1, 2.$$

Naturality thus holds when $\tilde{\kappa}_1 = \kappa_1$ and $\tilde{\kappa}_2 = \kappa_2$. \square

Remark 13. We can see in the proof of Lemma 12 why BFFG fails to be a lax monoidal functor in general: without the restriction $\tilde{\kappa} = \kappa$, the ‘natural’ choice of μ in fact fails to be a natural transformation: the w_i appearing in (4.8) can be interpreted as the importance weights picked up from the *other* branch of the computation, which in general, do *not equal* 1. While we do not formally prove that there do not exist any natural transformations that can make BFFG lax monoidal, we conjecture that no such transformations exist.

Lemma 14. Our morphisms μ satisfy the associativity condition.

Proof. We need to check that the associativity hexagon in Definition 15 holds.

This is somewhat tedious to verify, but it does hold: we need to check the actions on functions and measures, checking that for these mappings we can take either direction around the hexagon.

For functions, since pointwise products of functions is associative, both directions will map a triplet $(f(r), g(s), h(t))$ of functions to the pointwise product $(r, s, t) \mapsto f(r) \cdot g(s) \cdot h(t)$.

For measures, the marginalisation operation is also associative, in the sense that given a measure on a triple joint space $\bar{\mu}(dr, ds, dt)$, it does not matter in which order one performs the marginalisation: one can first form the intermediate pair of marginals $(\bar{\mu}(dr, S, T), \bar{\mu}(R, ds, dt))$ or the alternative pair $(\bar{\mu}(dr, ds, T), \bar{\mu}(R, S, dt))$. Given either of these intermediate pairs, further marginalisation will result in the same set of marginals, namely $(\bar{\mu}(dr, S, T), \bar{\mu}(R, ds, T), \mu(R, S, dt))$. \square

Lemma 15. Our morphisms ϵ, μ satisfy the unitality conditions.

Proof. We check now that the two unitality squares of Definition 15 hold.

For the left square, for functions we are comparing the identity mapping on a function g to the product $g \cdot 1$ with the constant 1 function. On measures, given a measure $\mu(ds)$, we first marginalise to obtain $(\mu(S) \cdot \delta_*, \mu)$ and then delete, resulting in $(*, \mu)$, as required.

The right square is essentially identical. \square

Remark 16. The previous two lemmas are valid in the general case of Stoch_2 and do not require the restriction to $\tilde{\kappa} = \kappa$.

Remark 17. Failure of lax-monoidality is not surprising. Clearly, upon first extracting marginals from the joint measure $\bar{\mu}$ and then applying BFFG on each of the marginals, one loses the dependency structure of $\bar{\mu}$, which is retained when applying BFFG on the joint space. However, when we have a product measure, $\bar{\mu}(ds, dt) = \bar{\mu}(ds, T)\bar{\mu}(S, dt)$, Equation (4.8) reads

$$\left(\int_S \frac{g_1(s')}{\tilde{\kappa}_1 g_1(s)} \bar{\mu}(ds, T) \kappa_1(s, ds') C_2, \int_T \frac{g_2(t')}{\tilde{\kappa}_2 g_2(t)} \bar{\mu}(S, dt) \kappa_2(t, dt') C_1 \right)$$

with C_1 and C_2 constants given by

$$C_1 = \int_S w_1(s) \bar{\mu}(ds, T) \quad \text{and} \quad C_2 = \int_T w_2(t) \bar{\mu}(S, dt).$$

This is to be compared to the expression for sm in (4.7). Algorithmically, since in applications of BFFG we typically initialise from a Dirac mass – which is indeed a product measure $\delta_{(x,y)} = \delta_x \otimes \delta_y$ – these constants are the only differences between the two sides of the naturality square. The output of the corresponding implementations of the BFFG algorithm is in both cases a sample from x from P° its likelihood, up to a constant factor. Considering the form of the final estimator T (2.14), these constants will ultimately cancel out, and hence are irrelevant. Thus, while lax-monoidality fails from a mathematical perspective, from a statistical perspective this failure is not an issue, and we can consider either implementation of BFFG.

Remark 18. We comment on links with the recent works of [Smithe \[2020, 2021\]](#), [Braithwaite et al. \[2023\]](#), which contain parallels of our results, when they are specialised to the BFFS setting. In these cited works, the general procedure of Bayesian inference (namely obtaining a posterior from a prior and data) is placed on a general categorical footing, and shown to have an optical

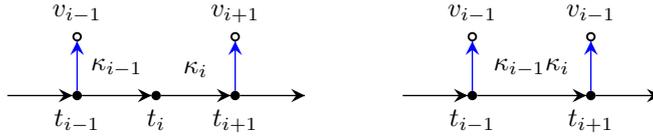


Figure 6: Visualisation for Examples 19 and 20. On the left, there are two separate kernels $\kappa_{i-1} : S_{i-1} \rightarrow S_i$ and $\kappa_i : S_i \rightarrow S_{i+1}$, with no observation at point t_i . On the right, there is a single (composed) kernel $\kappa_{i-1}\kappa_i : S_{i-1} \rightarrow S_{i+1}$.

structure, similar to our treatment of BFFG. Indeed [Smithe, 2020, Theorem 5.2] also demonstrates functoriality of Bayesian inversion into a category of optics; our Theorem 8 in the case of BFFS can thus be seen as a parallel of this result. For example, in the case of BFFS ($\tilde{\kappa} = \kappa$), we can dually interpret the function g as an (improper) ‘prior’ density, dually interpret the measure μ as a ‘test function’, and then the measure $\mathcal{F}_\kappa(g, \mu)$ from the second pass can be interpreted as a Bayesian inversion of the channel κ along ‘state’ g . Lax monoidality of the Bayesian inversion functor has also been established in [Braithwaite et al., 2023, Remark 22], under the assumption of the existence of *support objects* (for instance, in the case of finite state spaces or Gaussian kernels). This result, when applicable, thus parallels our Theorem 11, which we have proven in the setting of s -finite measures and measurable functions.

5 Examples

The results in this paper show that we may split an edge into multiple edges or, conversely, collapse multiple edges: the BFFG-algorithm is not affected by this. Below we briefly describe some examples that illustrate the relevance of this property.

Example 19 (Missing observation). Consider the hidden Markov model setting and suppose that at a hidden node t_i an observation is not recorded, perhaps due to a failing sensor – this situation is depicted on the left in Figure 6, where the observation v_i is absent. In this case, Theorem 8 tells us that we can collapse the edges (t_{i-1}, t_i) and (t_i, t_{i+1}) to (t_{i-1}, t_{i+1}) in the latent process, corresponding to the right diagram of Figure 6. Specifically, instead of performing two separate steps of BFFG with kernels κ_{i-1} and κ_i (and composing optics), we could instead perform a single step of BFFG with the composite kernel $\kappa_{i-1}\kappa_i$. Theorem 8 ensures that the overall algorithm is not affected by this.

Example 20 (Intractable composition). Consider the setting where a transition over an edge is captured by the kernel κ , which is intractable, but can be expressed as $\kappa = \kappa_1\kappa_2$ where both κ_1 and κ_2 are tractable. This suggests splitting the edge into two edges, where the added vertex in the middle represents a latent variable. Writing $\varphi(\mu, \sigma)$ for a Gaussian random variable with mean μ and covariance σ , a concrete example is $\kappa_1(x, dy) = \varphi(dy; \alpha x, \sigma^2)$ and $\kappa_2(x, dy) = \varphi(dy; \mu(x), \nu^2) dy$. In this case

$$\kappa(x, dy) = \int \kappa_1(x, dz)\kappa_2(z, dy) = \int \varphi(dz; \alpha x, \sigma^2)\varphi(dy; \mu(x), \nu^2).$$

Only for very specific choices of the map μ this can be computed in closed form, hampering the backward step. However, due to the structure of κ , we can split the backward step into the composition of two backward steps, where we take $\tilde{\kappa}_1 = \kappa_1$ and $\tilde{\kappa}_2(x, dy) = \varphi(dy; \tilde{\mu}, \sigma^2)$ for a user-specified value of $\tilde{\mu}$. Only the second of these steps involves an approximation. Diagrammatically, this corresponds with beginning from the *right* diagram of Figure 6, and introducing an auxiliary intermediary state, corresponding to the left diagram of Figure 6. Again, Theorem 8 allows us to conclude that BFFG remains the same when introducing this latent variable, performing two steps of BFFG with kernels κ_1, κ_2 respectively and then composing the optics.

Example 21 (SDE example). Consider an edge e . On this edge, the transition is governed by the evolution of the stochastic differential equation (SDE) $dX_t = \mu(X_t) dt + dW_t$ for $t \in [0, \tau_e]$.

Note that in general the transition densities for this SDE are not known in closed form. However, on short time intervals, an Euler discretisation provides an accurate approximation. As a result, one can include additional vertices that represent times $i\tau_e/n$, $i = 1, \dots, n-1$, for some (large) specified value of n . Similar to the previous example, the backward filtering step can now be split into n steps, where the approximate kernels $\tilde{\kappa}$ on vertices correspond to the Euler-discretised SDE.

Appendix

A Categorical definitions

We start with the definition of a category, as in [Pierce, 1991, (Definition 1.1.1)].

Definition 12. A category \mathcal{C} comprises of

1. a collection of objects;
2. a collection of arrows (also called morphisms);
3. operations assigning to each arrow f an object $\text{dom}f$, its domain, and an object $\text{cod}f$, its codomain, succinctly written $f: A \rightarrow B$ where $A = \text{dom}f, B = \text{cod}f$;
4. a composition operator assigning to each pair of arrows f and g with $\text{cod}f = \text{dom}g$, a composite arrow $g \circ f: \text{dom}f \rightarrow \text{cod}g$, satisfying the following associative law: for any $f: A \rightarrow B, g: B \rightarrow C$ and $h: C \rightarrow D$,

$$h \circ (g \circ f) = (h \circ g) \circ f;$$

5. for each object A , an identity arrow $\text{id}_A: A \rightarrow A$ satisfying the identity law: for any $f: A \rightarrow B, g: B \rightarrow A$,

$$f \circ \text{id}_A = f \text{ and } \text{id}_A \circ g = g.$$

Definition 13. Given two categories \mathcal{C}, \mathcal{D} a **functor** $F: \mathcal{C} \rightarrow \mathcal{D}$ is a map taking each \mathcal{C} -object A to a \mathcal{D} -object $F(A)$ and each \mathcal{C} -arrow $f: A \rightarrow B$ to a \mathcal{D} -arrow $F(f): F(A) \rightarrow F(B)$, such that

1. $F(g \circ f) = F(g) \circ F(f)$, whenever $g \circ f$ is well-defined;
2. $F(\text{id}_A) = \text{id}_{F(A)}$ for each object $A \in \mathcal{C}$.

The categories we are interested in possess an additional important structure: as well as sequential composition, as defined in Definition 12, they also possess a notion of *parallel* composition. This is formally captured in the notion of a *monoidal* category.

Definition 14. A **monoidal category** is a category \mathcal{C} equipped with the following:

1. \mathcal{C} is equipped with a functor $\otimes: \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$;
2. \mathcal{C} possesses a distinguished object $1 \in \mathcal{C}$;
3. there are isomorphisms $\alpha_{A,B,C}: (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C)$ for all objects A, B, C which are natural in A, B, C ;
4. for all objects $A \in \mathcal{C}$ there are isomorphisms, the *left-* and *right-unitors*, $\lambda_A: 1 \otimes A \rightarrow A$ and $\rho_A: A \otimes 1 \rightarrow A$, both natural in A ;

5. these morphisms are subject to the *pentagon equation*:

$$\begin{array}{ccc}
& ((A \otimes B) \otimes C) \otimes D & \\
\alpha_{A,B,C} \otimes \text{id}_D \swarrow & & \searrow \alpha_{A \otimes B, C, D} \\
(A \otimes (B \otimes C)) \otimes D & & (A \otimes B) \otimes (C \otimes D) \\
\alpha_{A, B \otimes C, D} \downarrow & & \downarrow \alpha_{A, B, C \otimes D} \\
A \otimes ((B \otimes C) \otimes D) & \xrightarrow{\text{id}_A \otimes \alpha_{B, C, D}} & A \otimes (B \otimes (C \otimes D))
\end{array}$$

6. and the *triangle equation*:

$$\begin{array}{ccc}
(A \otimes 1) \otimes B & \xrightarrow{\alpha_{A, 1, B}} & A \otimes (1 \otimes B) \\
& \searrow \rho_A \otimes 1_B & \swarrow 1_A \otimes \lambda_B \\
& & A \otimes B
\end{array} ;$$

7. the category is furthermore a **symmetric** monoidal category if it is equipped with natural isomorphisms $s_{A,B}: A \otimes B \rightarrow B \otimes A$ such that

$$s_{B,A} \circ s_{A,B} = \text{id}_{A \otimes B}$$

and the hexagon identity is satisfied:

$$\begin{array}{ccc}
(A \otimes B) \otimes C & \xrightarrow{\alpha_{A,B,C}} & A \otimes (B \otimes C) & \xrightarrow{s_{A,B \otimes C}} & (B \otimes C) \otimes A \\
s_{A,B} \otimes \text{id}_C \downarrow & & & & \downarrow \alpha_{B,C,A} \\
(B \otimes A) \otimes C & \xrightarrow{\alpha_{B,A,C}} & B \otimes (A \otimes C) & \xrightarrow{\text{id}_B \otimes s_{A,C}} & B \otimes (C \otimes A)
\end{array} .$$

Our key contribution will be describing how the BFFG algorithm *preserves both sequential and parallel composition*. This will be formalised by proving that the BFFG algorithm gives rise to a functor, which in some cases is lax monoidal.

Definition 15. Given two monoidal categories $(\mathcal{C}, \otimes_{\mathcal{C}}, 1_{\mathcal{C}}, \alpha^{\mathcal{C}}, \lambda^{\mathcal{C}}, \rho^{\mathcal{C}})$ and $(\mathcal{D}, \otimes_{\mathcal{D}}, 1_{\mathcal{D}}, \alpha^{\mathcal{D}}, \lambda^{\mathcal{D}}, \rho^{\mathcal{D}})$, a *lax monoidal functor* is a functor $F: \mathcal{C} \rightarrow \mathcal{D}$, together with

1. a morphism $\epsilon: 1_{\mathcal{D}} \rightarrow F(1_{\mathcal{C}})$;
2. for all objects $A, B \in \mathcal{C}$, morphisms $\mu_{A,B}: F(A) \otimes_{\mathcal{D}} F(B) \rightarrow F(A \otimes_{\mathcal{C}} B)$, natural in A, B :

$$\begin{array}{ccc}
F(A) \otimes_{\mathcal{D}} F(B) & \xrightarrow{\mu_{A,B}} & F(A \otimes_{\mathcal{C}} B) \\
F(f) \otimes_{\mathcal{D}} F(g) \downarrow & & \downarrow F(f \otimes_{\mathcal{C}} g) \\
F(C) \otimes_{\mathcal{D}} F(D) & \xrightarrow{\mu_{C,D}} & F(C \otimes_{\mathcal{C}} D)
\end{array} ;$$

3. these satisfy associativity: for all objects $X, Y, Z \in \mathcal{C}$, the following diagram commutes:

$$\begin{array}{ccc}
(F(A) \otimes_{\mathcal{D}} F(B)) \otimes_{\mathcal{D}} F(C) & \xrightarrow{\alpha_{F(A), F(B), F(C)}^{\mathcal{D}}} & F(A) \otimes_{\mathcal{D}} (F(B) \otimes_{\mathcal{D}} F(C)) \\
\mu_{A,B} \otimes \text{id} \downarrow & & \downarrow \text{id} \otimes \mu_{B,C} \\
F(A \otimes_{\mathcal{C}} B) \otimes_{\mathcal{D}} F(C) & & F(A) \otimes_{\mathcal{D}} F(B \otimes_{\mathcal{C}} C) \\
\mu_{A \otimes_{\mathcal{C}} B, C} \downarrow & & \downarrow \mu_{A, B \otimes_{\mathcal{C}} C} \\
F((A \otimes_{\mathcal{C}} B) \otimes_{\mathcal{C}} C) & \xrightarrow{F(\alpha_{A,B,C}^{\mathcal{C}})} & F(A \otimes_{\mathcal{C}} (B \otimes_{\mathcal{C}} C))
\end{array}$$

4. and satisfy unitality: for each object $A \in \mathcal{C}$, the following diagrams commute:

$$\begin{array}{ccc} 1_{\mathcal{D}} \otimes_{\mathcal{D}} F(A) & \xrightarrow{\epsilon \otimes \text{id}} & F(1_{\mathcal{C}}) \otimes_{\mathcal{D}} F(A) \\ \lambda_{F(A)}^{\mathcal{D}} \downarrow & & \downarrow \mu_{1_{\mathcal{C}}, A} \\ F(A) & \xleftarrow{F(\lambda_A^{\mathcal{C}})} & F(1_{\mathcal{C}} \otimes A) \end{array}$$

and

$$\begin{array}{ccc} F(A) \otimes_{\mathcal{D}} 1_{\mathcal{D}} & \xrightarrow{\text{id} \otimes \epsilon} & F(A) \otimes_{\mathcal{D}} F(1_{\mathcal{C}}) \\ \rho_{F(A)}^{\mathcal{D}} \downarrow & & \downarrow \mu_{A, 1_{\mathcal{C}}} \\ F(A) & \xleftarrow{F(\rho_A^{\mathcal{C}})} & F(A \otimes 1_{\mathcal{C}}) \end{array} .$$

References

- Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2007. ISBN 978-0-387-31073-2.
- Dylan Braithwaite, Jules Hedges, and Toby St. Clere Smithe. The Compositional Structure of Bayesian Inference. In *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, number 24 in Leibniz International Proceedings in Informatics, pages 1–24, 2023. ISBN 9783959772921. <https://doi.org/10.4230/LIPIcs.MFCS.2023.24>.
- Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in Hidden Markov Models*. New York: Springer, 2005.
- Chris K Carter and Robert Kohn. On Gibbs sampling for state space models. *Biometrika*, 81(3): 541–553, 1994. <https://doi.org/10.1093/biomet/81.3.541>.
- Kenta Cho and Bart Jacobs. Disintegration and Bayesian inversion via string diagrams. *Mathematical Structures in Computer Science*, 29(7):938–971, 2019. <https://doi.org/10.1017/S0960129518000488>.
- James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods*. Oxford University Press, 2nd edition, 2012. <https://doi.org/10.1093/acprof:oso/9780199641178.001.0001>.
- Brendan Fong. *Causal Theories: A Categorical Perspective on Bayesian Networks*. PhD thesis, University of Oxford, 2012.
- Brendan Fong, David Spivak, and Rémy Tuyéras. Backprop as functor: A compositional perspective on supervised learning. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13. IEEE, 2019. <https://doi.org/10.1109/LICS.2019.8785665>.
- Tobias Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, Aug 2020. <https://doi.org/10.1016/j.aim.2020.107239>.
- Tobias Fritz and Andreas Klingler. The d-separation criterion in Categorical Probability. *Journal of Machine Learning Research*, 24:1–49, 2023. <https://doi.org/10.48550/arXiv.2207.05740>.
- Tobias Fritz and Wendong Liang. Free gs-Monoidal Categories and Free Markov Categories. *Applied Categorical Structures*, 31(21):1–31, 2023. ISSN 15729095. <https://doi.org/10.1007/s10485-023-09717-0>.
- Tobias Fritz, Andreas Klingler, Drew McNeely, Areeb Shah-Mohammed, and Yuwen Wang. Hidden Markov Models and the Bayes Filter in Categorical Probability. *arXiv preprint arXiv:2401.14669*, 2024. <https://doi.org/10.55776/P35992>.
- Sylvia Frühwirth-Schnatter. Data augmentation and dynamic linear models. *Journal of time series analysis*, 15(2):183–202, 1994. <https://doi.org/10.1111/j.1467-9892.1994.tb00186.x>.
- Bruno Gavranović. Space-time tradeoffs of lenses and optics via higher category theory. *arXiv preprint arXiv:2209.09351*, 2022. <https://doi.org/10.48550/arXiv.2209.09351>.
- Bart Jacobs, Aleks Kissinger, and Fabio Zanasi. Causal Inference by String Diagram Surgery. In M. Bojańczyk and A Simpson, editors, *Foundations of Software Science and Computation Structures. FoSSaCS 2019*, volume 11425 of *Lecture Notes in Computer Science*, pages 313–329. Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-17127-8_18.

- Marcin Mider, Moritz Schauer, and Frank Van der Meulen. Continuous-discrete smoothing of diffusions. *Electronic Journal of Statistics*, 15(2):4295–4342, 2021. <https://doi.org/10.1214/21-EJS1894>.
- Benjamin C Pierce. *Basic category theory for computer scientists*. MIT press, 1991.
- Lawrence R. Rabiner and Biing-Hwang Juang. *Theory and Applications of Hidden Markov Models*. Springer, 2007.
- Mitchell Riley. Categories of optics. *arXiv preprint arXiv:1809.00738*, 2018. <https://doi.org/10.48550/arXiv.1809.00738>.
- Dan Shiebler, Bruno Gavranović, and Paul Wilson. Category theory in machine learning. *arXiv preprint arXiv:2106.07032*, 2021. <https://doi.org/10.48550/arXiv.2106.07032>.
- Toby St. Clere Smithe. Bayesian Updates Compose Optically. *arXiv preprint arXiv:2006.01631*, 2020. <https://doi.org/10.48550/arXiv.2006.01631>.
- Toby St. Clere Smithe. Compositional Active Inference I: Bayesian Lenses. *Statistical Games*. *arXiv preprint arXiv:2109.04461*, 2021. <https://doi.org/10.48550/arXiv.2109.04461>.
- Sam Staton. Commutative semantics for probabilistic programming. In H Yang, editor, *Programming Languages and Systems*, volume 10201 of *Lecture Notes in Computer Science*, pages 855–879. Springer, Berlin, Heidelberg, 2017. https://doi.org/10.1007/978-3-662-54434-1_32.
- Frank Van der Meulen. Introduction to Automatic Backward Filtering Forward Guiding. *arXiv preprint arXiv:2203.04155*, 2022. <https://doi.org/10.48550/arXiv.2203.04155>.
- Frank Van der Meulen and Moritz Schauer. Automatic backward filtering forward guiding for Markov processes and graphical models. *arXiv preprint arXiv:2010.03509*, 2022. <https://doi.org/10.48550/arXiv.2010.03509>.
- Pietro Vertechi. Dependent optics. *arXiv preprint arXiv:2204.09547*, 2022. <https://doi.org/10.48550/arxiv.2204.09547>.
- Nathaniel Virgo. Unifilar Machines and the Adjoint Structure of Bayesian Filtering. *Electronic Proceedings in Theoretical Computer Science, EPTCS*, 397:299–317, 2023. ISSN 20752180. <https://doi.org/10.4204/EPTCS.397.16>.
- Mike West and Jeff Harrison. *Bayesian Forecasting and Dynamic Models*. Springer, 2nd edition, 1997.