Solving the Filtering Equation

Danilo Jr Dela Cruz
Mathematics of Random Systems
University of Oxford

21st January 2024

Abstract

Stochastic Filtering is a technique which enables the integration of observations and an understanding of the dynamics to obtain improved estimates of a latent signal process. We present an exposition of Crisan et al. [2022], which provides a method for solving the filtering equation. Through a series of reductions, Crisan et al. [2022] derive a Feynman-Kac inspired loss function for neural networks to solve the PDE induced by the splitting-up method on the associated Zakai SPDE.

1 Introduction

Given a filtered probability space $(\Omega, \mathcal{F}, \mathcal{F}_t, \mathbb{P})$, we have a signal process $X_t \in \mathbb{R}^d$ and observation process $Y_t \in \mathbb{R}^m$, modelling noise via independent standard brownian motions $V_t \in \mathbb{R}^p, W_t \in \mathbb{R}^m$.

$$dX_t = f(X_t)dt + \sigma(X_t)dV_t \tag{1}$$
$$dY_t = h(X_t)dt + dW_t \tag{2}$$

As a technical condition, we assume X_0 has finite third moment, f, σ are (globally) Lipschitz and the sensor function h satisfies a linear growth condition. See Bain and Crisan [2009] for further details.

The goal of stochastic filtering is to obtain the filter π_t which represents the best estimate for the latent signal process X_t given our observations Y_t.

$$\pi_t \varphi = \mathbb{E}\left[\varphi(X_t) | \mathcal{Y}_t\right] \tag{3}$$

where $\mathcal{Y}_t = \sigma\left(Y_s : s \in [0,t]\right) \vee \mathcal{N}$ is the observation filtration and $\mathcal{N}$ are the $\mathbb{P}$-nullsets of $\mathcal{F}$.

It is convenient to consider the change of measure $\mathbb{Q}$ and the unnormalised filter ρ_t

$$\left.\frac{d\mathbb{Q}}{d\mathbb{P}}\right|_{\mathcal{F}_t} = Z_t = \mathcal{E}\left(\int_0^{\cdot} h(X_s)^T dW_s\right)_t \tag{4}$$

$$\rho_t \varphi = \mathbb{E}_{\mathbb{Q}}\left[\varphi(X_t) Z_t^{-1} \middle| \mathcal{Y}_t\right]. \tag{5}$$

This is because

- Y_t is a standard brownian motion wrt $(\mathcal{F}_t, \mathbb{Q})$
- Hence $\mathcal{Y}_t$ is a brownian filtration which allows the unnormalised filter
$$\rho_t = \mathbb{E}_{\mathbb{Q}}\left[\varphi(X_t) Z_t^{-1} \middle| \mathcal{Y}\right] \tag{6}$$
to be represented with a time-independent filtration $\mathcal{Y} = \lim_{t\to\infty} \mathcal{Y}_t$.
- The SPDE 9 for ρ_t is simpler than that of π_t. (See Theorem 3.30 in Bain and Crisan [2009] for the Kushner-Stratonovich Equation for π_t.)

Furthermore, we can recover the normalised filter via the Kallianpur-Striebel Formula

$$\pi_t \varphi = \frac{\rho_t \varphi}{\rho_t 1}. \tag{7}$$

1.1 Reduction to solving an SPDE

Under suitable conditions given by Theorem 7.8 and 7.12 Bain and Crisan [2009], ρ_t admits a density p_t in the sense that

$$\rho_t(\varphi) = \int_{\mathbb{R}^d} \varphi(x) p_t(x) dx. \tag{8}$$

$p_t(x)$ is a random density as it depends on $\mathcal{Y}_t$, but it can be treated deterministically after observation of Y_t. The density satisfies a Zakai Equation

$$p_t(x) = p_0(x) + \int_0^t A^\star p_s(x) ds + \int_0^t p_s(x) h(x)^T dY_s. \tag{9}$$

where $A^\star$ is the adjoint of the markov generator A of the signal process X_t.

1.2 Splitting-up Method

In order to solve equation 9, we consider the splitting-up method proposed in LeGland [1992] which returns an approximation to p_t given an initial condition p_0. The method solves each term in the SPDE separately (it has "split-up" the equation). The two phases are analogous to the prediction-correction paradigm in filtering.

The quality of approximation increases for smaller time steps and to solve over $[0, T]$ we can take multiple small steps. Formally, we consider a mesh $\{0 = t_0, \cdots, t_N = T\}$ and chain the splitting-up method to return a collection of functions $\widehat{p}_n(x) \approx p_{t_n}(x)$. We let $\widehat{p}_0(x) = p_0(x)$ and to chain we set $\widehat{p}_{n-1}(x)$ as the initial condition for $\widehat{p}_n(x)$.

Phase 1: Prediction
Solve for $\widetilde{p}_n(t, x)$ over $(t_{n-1}, t_n] \times \mathbb{R}^d$

$$\frac{\partial \widetilde{p}_n}{\partial t}(t, z) = A^\star \widetilde{p}_n(t, z) \tag{10}$$
$$\widetilde{p}_n(t_{n-1}, z) = \widehat{p}_{n-1}(z). \tag{11}$$

Phase 2: Correction
Compute

$$z_n = \frac{Y_{t_n} - Y_{t_{n-1}}}{t_n - t_{n-1}} \approx \frac{dY(t_n)}{dt}. \tag{12}$$

Generate the correction multiplier

$$\xi_n(x) = \exp\left(-\frac{t_n - t_{n-1}}{2} \|z_n - h(x)\|^2\right) \tag{13}$$

Normalise as $\widehat{p}_n \approx p_{t_n}$ approximates a density

$$\widehat{p}_n(x) = \frac{\xi(x)\widetilde{p}_n(t_n, x)}{\int_{\mathbb{R}^d} \xi(z)\widetilde{p}_n(t_n, z)dz}. \tag{14}$$

1.3 Neural Network PDE Solvers

To solve the prediction step PDE, we use a neural network $\mathcal{NN}_\theta$ trained on a Feynman-Kac inspired loss function given in Beck et al. [2021]. Due to the lack of boundary conditions, this method has the advantage that it does not require boundary conditions. Instead, we can choose a fixed space domain $[a, b]^d$ and train the neural network to match the Feynman-Kac solution over the specified domain.

$$\mathcal{L}_{\text{FK}}(\theta) = \mathbb{E}\left[\left|\psi(\widehat{X}_T)S_T - \mathcal{NN}_\theta(\xi)\right|^2\right] \tag{15}$$
$$S_T = \exp\left(\int_0^T r(\widehat{X}_t)dt\right) \tag{16}$$

Here $\widehat{X}_t$ is the auxiliary diffusion defined in Equation 37 and $\xi \sim \text{Unif}([a, b]^d)$.

In the rest of this article, we will discuss the intuition behind the Splitting-up Method and the Feynman-Kac inspired loss function.

- Section 2 places the Splitting-up method proposed by LeGland [1992] in the context of Operator Splitting Methods.
- Section 3 shows how we arrive at the Feynman-Kac inspired loss function.

2 Splitting-up Method

Given a PDE operator which is a sum of time-independent linear operators, operator splitting methods obtain an approximation by solving one-term-at-a-time and combining the multiple solutions Yazici [2010].

The Splitting-up Method from LeGland [1992] can be seen as an extension of the Lie-Trotter scheme

Definition 2.1 (Lie-Trotter Scheme)**.** *Let A, B be operators in space and consider the ODE given by.*

$$\frac{\partial u}{\partial t} = A \circ u + B \circ u, \quad u(0) = u_0 \tag{17}$$

We will form a solution by solving for w, v over $[0, T]$. And our approximation is $u(T) \approx v(T)$.

$$\frac{\partial w}{\partial t} = A \circ w, \quad w(0) = u_0 \tag{18}$$
$$\frac{\partial v}{\partial t} = B \circ v, \quad v(0) \ = w(T) \tag{19}$$

The solution is $\exp(T(A + B)) \circ u_0$ but the Lie-Trotter Scheme returns $\exp(TB) \circ \exp(TA) \circ u_0$.

$$w(T) = \exp(TA) \circ u_0 \tag{20}$$
$$v(T) = \exp(TB) \circ w(T) \tag{21}$$
$$= \exp(TB) \circ \exp(TA) \circ u_0 \tag{22}$$

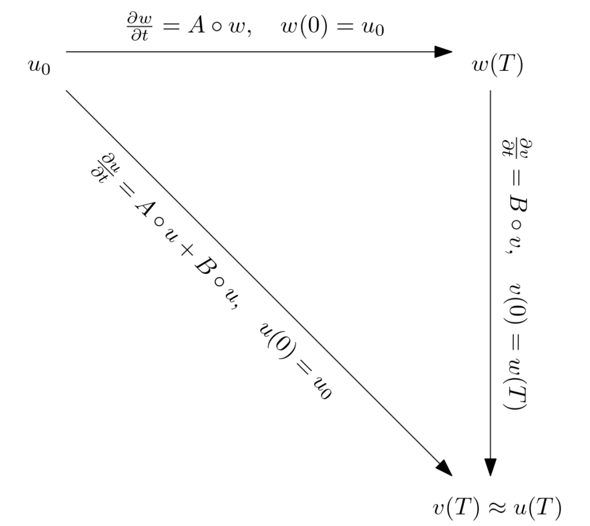

Figure 1: Illustration of Lie Trotter Scheme which takes in initial condition and outputs terminal value approximant $v(T)$. (Made with Ipe.)

The approximation is only exact when A, B commute, but in general the error should decrease for smaller T. By the Baker-Campbell-Hausdorff Formula $\exp(TB) \circ \exp(TA) \approx \exp(T(B + A))$ as

$$\exp(TB) \circ \exp(TA) = \exp(C). \tag{23}$$

C is dominated by $T(B+A)$ and higher order terms consist of Lie Brackets $[A, B] = AB - BA$.

$$C = T(B + A) + \frac{T^2}{2}[B, A] + O\left(T^3\right) \tag{24}$$

In fact the Lie Trotter Scheme is first order, see Section 2.1 in Yazici [2010].

$$\frac{u(T) - v(T)}{T} = \frac{T}{2}[A, B] \circ u_0 + O\left(T^2\right) \tag{25}$$

2.1 Zakai Equation

In our setting, it is slightly more complicated as we have an SPDE and one of the operators depend on time. But, the idea is still the same.

$$dp(t, x) = \underbrace{A^\star}_{A} p(t, x)dt + \underbrace{h(x)^T dY_t}_{B} p(t, x) \tag{26}$$

Under the Lie Trotter scheme, we first solve the PDE yielding $w(T, x) = \tilde{p}(T, x)$. The second step yields a Stochastic Exponential SDE.

$$dv(t, x) = v(t, x)h(x)^T dY_t, \quad v(0, x) = \tilde{p}(T, x)$$
$$v(T, x) = \tilde{p}(T, x)\mathcal{E}\left(\int_0^{\cdot} h(x)^T dY_s\right)$$

Letting $z = Y_T - Y_0/T$ and simplifying yields

$$\frac{v(T, x)}{\tilde{p}(T, x)} = \exp\left(h(x)^T(Y_T - Y_0) - \frac{T}{2}\|h(x)\|^2\right) \tag{27}$$
$$= \exp\left(-\frac{T}{2}\|h(x) - z\|^2 + \frac{T}{2}\|z\|^2\right) \tag{28}$$
$$= \xi(x) \exp\left(\frac{T}{2}\|z\|^2\right). \tag{29}$$

As the final step is normalise, we can omit $\exp\left(\frac{T}{2}\|z\|^2\right)$ which allows $\int \tilde{p}(T, y)\xi(y)dy$ to be interpreted as an expectation wrt a normal distribution restricted to the space domain.

2.2 Rate of convergence

Gyöngy and Krylov [2003] demonstrate that the rate of convergence is $O(1/n)$ when the time steps are of size T/n. The norm is taken over the Sobolev Space $S = W_2^m(\mathbb{R}^d)$ and valid for any $m \geq 0$.

$$\mathbb{E}\left[\max_{t \in T_n}\left\|p^{(n)}(t, x) - p(t, x)\right\|_S^2\right]^{1/2} \leq C/n \tag{30}$$

3 Solving the PDE

The Splitting-up method requires us to solve PDEs with the PDE operator being the adjoint markov generator $A^\star$ and initial condition ψ over $(0, t]$.

$$\frac{\partial u}{\partial t}(t, x) = A^\star u(t, x) \tag{31}$$
$$u(0, x) = \psi(x) \tag{32}$$

We first show how it can be reformulated for Feynman-Kac. First, for a given diffusion

$$dX_t = f(X_t)dt + \sigma(X_t)dV_t, \tag{33}$$

it's generator and dual generator can be written as

$$A\varphi = \langle f, \nabla\varphi\rangle + \mathrm{Tr}\,(a \text{ Hess}\,\varphi) \tag{34}$$
$$A^\star\varphi = \underbrace{\langle b, \nabla\varphi\rangle + \mathrm{Tr}\,(a \text{ Hess}\,\varphi)}_{\widehat{A}\varphi} + r\varphi \tag{35}$$

where $a = \frac{1}{2}\sigma\sigma^T, b = 2\overrightarrow{\mathrm{div}}(a) - f$ and $r = \mathrm{div}\left(\overrightarrow{\mathrm{div}}(a) - f\right)$. $\overrightarrow{\mathrm{div}}(\cdot)$ corresponds to the vector obtained by taking the divergence of each column of a matrix-valued function.

$$\mathrm{div}(f) = \sum_{i=1}^{d} \partial_i f, \quad \overrightarrow{\mathrm{div}}(a)_j = \mathrm{div}(a_{\cdot j}) \tag{36}$$

$A^\star$ does not correspond to a diffusion however it can be decomposed into $A^\star = \widehat{A} + r$ where $\widehat{A}$ corresponds to a diffusion $\widehat{X}_t$.

$$d\widehat{X}_t = b(\widehat{X}_t)dt + \sigma(\widehat{X}_t)dV_t \tag{37}$$

Now we can apply Feynman-Kac to obtain a loss function, see Corollary 2.3 in Crisan et al. [2022].

Theorem 3.1. *Under suitable assumptions, the solution of the following PDE*

$$\frac{\partial u}{\partial t}(t,x) = \widehat{A}u(t,x) + r(x)u(t,x) \tag{38}$$

$$u(0,x) = \psi(x) \tag{39}$$

has stochastic representation in terms of the diffusion $\widehat{X}_t$ generated by $\widehat{A}$.

$$u(t,x) = \mathbb{E}\left[\psi(\widehat{X}_t)S_t \,\middle|\, \widehat{X}_0 = x\right] \tag{40}$$

$$S_t = \exp\left(\int_0^t r(\widehat{X}_s)ds\right) \tag{41}$$

3.1 Loss Function

The expectation $\mathbb{E}[X]$ corresponds to the value μ which minimises the mse $\mathbb{E}\left[|X-\mu|^2\right]$. Proposition 2.4 in Crisan et al. [2022] extends this analogy.

Theorem 3.2. *Consider the PDE and stochastic representation $u(t,x)$ given in Theorem 3.1. For any $a < b$, and $D \sim \mathrm{Unif}([a,b]^d)$, $U(x) = u(T,x)$ is the unique continuous function which minimises $\mathcal{L}(v)$ over all continuous function $v \in C([a,b]^d, \mathbb{R})$.*

$$\mathcal{L}(v) = \mathbb{E}\left[\left|\psi(\widehat{X}_t)S_t - v(D)\right|^2\right] \tag{42}$$

$\widehat{X}_t$ is the diffusion generated by $\widehat{A}$ with initial distribution $\widehat{X}_0 = D$.

Now all that remains is to let $v = \mathcal{NN}_\theta$ be a neural network and minimise this loss function to obtain an approximation of the PDE solution.

References

Alan Bain and Dan Crisan. *Fundamentals of Stochastic Filtering*, volume 60 of *Stochastic Modelling and Applied Probability*. Springer New York, New York, NY, 2009. ISBN 978-0-387-76895-3 978-0-387-76896-0. doi: 10.1007/978-0-387-76896-0. URL http://link.springer.com/10.1007/978-0-387-76896-0.

Christian Beck, Sebastian Becker, Philipp Grohs, Nor Jaafari, and Arnulf Jentzen. Solving the Kolmogorov PDE by means of deep learning. *Journal of Scientific Computing*, 88(3):73, September 2021. ISSN 0885-7474, 1573-7691. doi: 10.1007/s10915-021-01590-0. URL http://arxiv.org/abs/1806.00421. arXiv:1806.00421 [cs, math, stat].

Dan Crisan, Alexander Lobbe, and Salvador Ortiz-Latorre. An application of the splitting-up method for the computation of a neural network representation for the solution for the filtering equations, January 2022.

István Gyöngy and Nicolai Krylov. On the Rate of Convergence of Splitting-up Approximations for SPDEs. In Evariste Giné, Christian Houdré, and David Nualart, editors, *Stochastic Inequalities and Applications*, pages 301–321. Birkhäuser Basel, Basel, 2003. ISBN 978-3-0348-9428-9 978-3-0348-8069-5. doi: 10.1007/978-3-0348-8069-5_17.

François LeGland. Splitting-up approximation for SPDE's and SDE's with application to nonlinear filtering. In Boris L. Rozovskii and Richard B. Sowers, editors, *Stochastic Partial Differential Equations and Their Applications*, volume 176, pages 177–187. Springer-Verlag, Berlin/Heidelberg, 1992. ISBN 978-3-540-55292-5. doi: 10.1007/BFb0007332. URL http://link.springer.com/10.1007/BFb0007332. Series Title: Lecture Notes in Control and Information Sciences.

Yesim Yazici. Operator splitting methods for differential equations, May 2010.