Learning Stochastic Dynamical Systems via Bridge Sampling

Anonymous Author(s)
Affiliation
Address
email

Abstract

The abstract paragraph should be indented ½ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

The goal of this work is to enable automatic discovery of stochastic differential equations (SDE) from time series data.

1. Literature review. simple, effective procedure for ODE (not SDE): [3–5] low-dimensional direct approach with gradient-based optimization (not scalable to higher dim): [1] diffusion bridges but more complex estimation (not EM!): [6, 17] classical stat approaches: [7, 8, 18] PDE: [12–14] compressed sensing: [15, 16]
2. What is new and interesting about this work.

Points to cover:

- data specification, DONE
- Hermite polynomial and drift function representation, DONE
- Expectation and maximization formulas assuming data is filled in, CLOSE2DONE
- Filling data in with diffusion bridge, DONE
- MCMC iterations of brownian bridge using girsanov likelihood, DONE
- how synthetic data is generated
- results: 1D, 2D, 3D damped duffing, 3D lorenz
- plots: error of theta vs noise, error vs amount of data (number of data points) parametric curves for noise levels, brownian bridge plots for illustration, ...

1 Problem Setup

Let W_t denote Brownian motion in $\mathbb{R}^d$—informally, an increment dW_t of this process has a multivariate normal distribution with zero mean vector and covariance matrix $I dt$. Let X_t denote an $\mathbb{R}^d$-valued stochastic process that evolves according to the Itô SDE

$$dX_t = f(X_t)dt + \Gamma dW_t. \tag{1}$$

For rigorous definitions of Brownian motion and SDE, see [2, 19]. The nonlinear vector field $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}^d$ is the *drift* function, and the $d \times d$ matrix Γ is the *diffusion* matrix. To reduce the number of model parameters, we assume $\Gamma = \operatorname{diag} \gamma$.

Our goal is to develop an algorithm that accurately estimates the functional form of f and the vector γ from time series data.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

Parameterization. We parameterize f using Hermite polynomials. The n-th Hermite polynomial takes the form

$$H_n(x) = (\sqrt{2\pi}n!)^{-1/2}(-1)^n e^{x^2/2} \frac{\mathrm{d}^n}{\mathrm{d}x^n} e^{-x^2/2} \tag{2}$$

Let $\langle f, g\rangle_w = \int_{\mathbb{R}} f(x)g(x)\exp(-x^2/2)\,dx$ denote a weighted L^2 inner product. Then, $\langle H_i, H_j\rangle_w = \delta_{ij}$, i.e., the Hermite polynomials are orthonormal with respect to the weighted inner product. In fact, with respect to this inner product, the Hermite polynomials form an orthonormal basis of $L^2_w(\mathbb{R}) = \{f : \langle f, f\rangle_w < \infty\}$.

Now let $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{Z}^d_+$ denote a multi-index. We use the notation $|\alpha| = \sum_j \alpha_j$ and $x^\alpha = \prod_j (x_j)^{\alpha_j}$ for $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$. For $x \in \mathbb{R}^d$ and a multi-index α, we also define

$$H_\alpha(x) = \prod_{j=1}^{d} H_{\alpha_j}(x_j). \tag{3}$$

We write $f(x) = (f_1(x), \ldots f_d(x))$ and then parameterize each component:

$$f_j(x) = \sum_{m=0}^{M} \sum_{|\alpha|=m} \beta^j_\alpha H_\alpha(x). \tag{4}$$

We see that the maximum degree of $H_\alpha(x)$ is $|\alpha|$. Hence we think of the double sum in (4) as first summing over degrees and then summing over all terms with a fixed maximum degree. We say maximum degree because, for instance, $H_2(z) = (z^2 - 1)/(\sqrt{2\pi}2)^{1/2}$ contains both degree 2 and degree 0 terms.

There are $\binom{m+d-1}{d-1}$ possibilities for a d-dimensional multi-index α such that $|\alpha| = m$. Summing this from $m = 0$ to M, there are $\widetilde{M} = \binom{M+d}{d}$ total multi-indices in the double sum in (4). Let (i) denote the i-th multi-index according to some ordering. Then we can write

$$f_j(x) = \sum_{i=1}^{\widetilde{M}} \beta^j_{(i)} H_{(i)}(x). \tag{5}$$

Data. We consider our data $\mathbf{x} = \{x_j\}_{j=0}^L$ to be direct observations of X_t at discrete points in time $\mathbf{t} = \{t_j\}_{t=0}^L$. Note that these time points do not need to be equispaced.

To achieve our estimation goal, we apply expectation maximization (EM). We regard $\mathbf{x}$ as the incomplete data. Let $\Delta t = \max_j (t_j - t_{j-1})$ be the maximum interobservation spacing. We think of the missing data $\mathbf{z}$ as data collected at a time scale $h \ll \Delta t$ that is fine enough such that the transition density of (1) is approximately Gaussian. To see how this works, let $\mathcal{N}(\mu, \Sigma)$ denote a multivariate normal with mean vector μ and covariance matrix Σ. Now discretize (1) in time via the Euler-Maruyama method with time step $h > 0$; the result is

$$\widetilde{X}_{n+1} = \widetilde{X}_n + f(\widetilde{X}_n)h + h^{1/2}\Gamma Z_{n+1}, \tag{6}$$

where $Z_{n+1} \sim \mathcal{N}(0, I)$ is a standard multivariate normal, independent of X_n. This implies that

$$(\widetilde{X}_{n+1} | \widetilde{X}_n = v) \sim \mathcal{N}(v + f(v)h, h\Gamma^2). \tag{7}$$

As h decreases, $\widetilde{X}_{n+1}|\widetilde{X}_n = v$—a Gaussian approximation—will converge to the true transition density $X_{(n+1)h}|X_{nh} = v$, where X_t refers to the solution of (1).

Diffusion Bridge. To augment or complete the data, we employ diffusion bridge sampling, using a Markov chain Monte Carlo (MCMC) method that goes back to [10, 11]. Let us describe our version here. We suppose our current estimate of $\boldsymbol{\theta} = (\beta, \gamma)$ is given. Define the diffusion bridge process to be (1) conditioned on both the initial value x_i at time t_i, and the final value x_{i+1} at time t_{i+1}. The goal is to generate sample paths of this diffusion bridge. By a sample path, we mean $F - 1$ *new* samples $\{z_{i,j}\}_{j=1}^{F-1}$ at times $t_i + jh$ with $h = (t_{i+1} - t_i)/F$.

To generate such a path, we start by drawing a sample from a Brownian bridge with the same diffusion as (1). That is, we sample from the SDE

$$d\widehat{X}_t = \Gamma dW_t \tag{8}$$

conditioned on $\widehat{X}_{t_i} = x_i$ and $\widehat{X}_{t_{i+1}} = x_{i+1}$. This Brownian bridge can be described explicitly:

$$\widehat{X}_t = \Gamma(W_t - W_{t_i}) + x_i - \frac{t - t_i}{t_{i+1} - t_i}(\Gamma(W_{t_{i+1}} - W_{t_i}) + x_i - x_{i+1}) \tag{9}$$

Here $W_0 = 0$ (almost surely), and $W_t - W_s \sim \mathcal{N}(0, (t-s)I)$ for $t > s \geq 0$.

Let $\mathbb{P}$ denote the law of the diffusion bridge process, and let $\mathbb{Q}$ denote the law of the Brownian bridge (9). Using Girsanov's theorem [9], we can show that

$$\frac{d\mathbb{P}}{d\mathbb{Q}} = C \exp\left(\int_{t_i}^{t_{i+1}} f(\widehat{X}_s)^T \Gamma^{-2} \, d\widehat{X}_s - \frac{1}{2} \int_{t_i}^{t_{i+1}} f(\widehat{X_s})^T \Gamma^{-2} f(\widehat{X_s}) \, ds \right), \tag{10}$$

where the constant C depends only on x_i and x_{i+1}. The left-hand side is a Radon-Nikodym derivative, equivalent to a density or likelihood; the ratio of two such likelihoods is the accept/reject ratio in the Metropolis algorithm [Stuart 2010].

Putting the above pieces together yields the following Metropolis algorithm to generate diffusion bridge sample paths. Fix $F \geq 2$ and $i \in \{0, \ldots, L-1\}$. Assume we have stored the previous Metropolis step, i.e., a path $\mathbf{z}^{(\ell)} = \{z_{i,j}\}_{j=1}^{F-1}$.

1. Use (9) to generate samples of $\widehat{X}_t$ at times $t_i + jh$, for $j = 1, 2, \ldots, F-1$ and $h = (t_{i+1} - t_i)/F$. This is the proposal $\mathbf{z}^* = \{z^*_{i,j}\}_{j=1}^{F-1}$.

2. Numerically approximate the integrals in (10) to compute the likelihood of the proposal. Specifically, we compute

$$\begin{aligned} p(\mathbf{z}^*)/C = &\sum_{j=0}^{F-1} f(z^*_{i,j})^T \Gamma^{-2} (z^*_{i,j+1} - z^*_{i,j}) \\ &- \frac{h}{4} \sum_{j=0}^{F-1} \left[f(z^*_{i,j})^T \Gamma^{-2} f(z^*_{i,j}) + f(z^*_{i,j+1})^T \Gamma^{-2} f(z^*_{i,j+1}) \right] \end{aligned}$$

3. Accept the proposal with probability $p(\mathbf{z}^*)/p(\mathbf{z}^{(\ell)})$—note the factors of C cancel. If the proposal is accepted, then set $\mathbf{z}^{(\ell+1)} = \mathbf{z}^*$. Else set $\mathbf{z}^{(\ell+1)} = \mathbf{z}^{(\ell)}$.

We initialize this algorithm with a Brownian bridge path, run for 10 burn-in steps, and then use subsequent steps as the diffusion bridge samples we seek.

Expectation Maximization (EM). Let us now give details to justify the intuition expressed above, that employing the diffusion bridge to augment the data on a fine scale will enable estimation. Let $\mathbf{z}$ a diffusion bridge sample path. We interleave this sampled data together with the observed data $\mathbf{x}$ to create the completed time series

$$\mathbf{y}^{(r)} = \{y_j^{(r)}\}_{j=1}^N,$$

where $N = LF + 1$. By interleaving, we mean that $y^{(r)}_{1+iF} = x_i$ for $i = 0, 1, \ldots, L$, and that $y^{(r)}_{1+j+iF} = z_{i,j}$ for $j = 1, 2, \ldots, F-1$ and $i = 0, 1, \ldots, L-1$.

Let h_j denote the elapsed time between observations y_j and y_{j+1}. Using the completed data, the temporal discretization (6) of the SDE, the Markov property, and property (7), we have:

$$
\begin{aligned}
\log p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta}) &= \log p(\mathbf{y} \mid \boldsymbol{\theta}) \\
&= \sum_{j=1}^{N-1} \log p(y_{j+1} \mid y_j, \boldsymbol{\theta}) \\
&= -\sum_{j=1}^{N-1} \Bigg[\sum_{i=1}^{d} \frac{1}{2} \log(2\pi h_j \gamma_i^2) \\
&+ \frac{1}{2h_j} (y_{j+1} - y_j - h_j \sum_{k=1}^{\widetilde{M}} \beta_{(k)} H_{(k)}(y_j))^T \Gamma^{-2} (y_{j+1} - y_j - h_j \sum_{\ell=1}^{\widetilde{M}} \beta_{(\ell)} H_{(\ell)}(y_j)) \Bigg].
\end{aligned}
$$

EM. The EM algorithm consists of two steps, computing the expectation of the log likelihood function (on the completed data) and then maximizing it with respect to the parameters

1. Start with an initial guess for the parameters, $\boldsymbol{\theta}^{(0)}$.
2. For the expectation (or E) step,
$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) = \mathbb{E}_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}^{(k)}}[\log p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})] \tag{11}$$
Our plan is to evaluate this expectation via bridge sampling. That is, we will sample from diffusion bridges $\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta}^{(k)}$. Then $(\mathbf{x}, \mathbf{z})$ will be a combination of the original data together with sample paths.
3. For the maximization (or M) step, we start with the current iterate and a dummy variable $\boldsymbol{\theta}$ and define
$$\boldsymbol{\theta}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) \tag{12}$$
It will turn out that we can maximize this quantity without numerical optimization. All we will need to do is solve a least-squares problem.
4. Iterate Step 2 and 3 until convergence.

Details. With a fixed parameter vector $\boldsymbol{\theta}^{(k)}$, the SDE (1) is specified completely, i.e., the drift and diffusion terms have no further unknowns.

Suppose we form R such time series. The expected log likelihood can then be approximated by

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) &= \mathbb{E}_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}^{(k)}}[\log p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})] \\
&\approx \frac{1}{R} \sum_{r=1}^{R} \Bigg[\sum_{j=1}^{N} \left[\sum_{i=1}^{d} -\frac{1}{2} \log(2\pi h \gamma_i^2) \right] \\
&\quad - \frac{1}{2h} (y_j^{(r)} - y_{j-1}^{(r)} - h \sum_{k=1}^{M} \beta_k \phi_k(y_{j-1}^{(r)}))^T \Gamma^{-2} (y_j^{(r)} - y_{j-1}^{(r)} - h \sum_{\ell=1}^{M} \beta_\ell \phi_\ell(y_{j-1}^{(r)})) \Bigg]
\end{aligned}
$$

To maximize Q over $\boldsymbol{\theta}$, we first assume $\Gamma = \operatorname{diag} \gamma$ is known and maximize over β. This is a least squares problem. The solution is given by forming the matrix

$$\mathcal{M}_{k,\ell} = \frac{1}{R} \sum_{r=1}^{R} \sum_{j=1}^{N} h \phi_k^T(y_{j-1}^{(r)}) \Gamma^{-2} \phi_\ell^T(y_{j-1}^{(r)})$$

and the vector

$$\rho_k = \frac{1}{R} \sum_{r=1}^{R} \sum_{j=1}^{N} \phi_k^T(y_{j-1}^{(r)}) \Gamma^{-2} (y_j^{(r)} - y_{j-1}^{(r)}).$$

We then solve the system

$$\mathcal{M}\beta = \rho$$

for β. Now that we have β, we maximize Q over γ. The solution can be obtained in closed form:

$$\gamma_i^2 = \frac{1}{RNh}\sum_{r=1}^{R}\sum_{j=1}^{N}((y_j^{(r)} - y_{j-1}^{(r)} - h\sum_{\ell=1}^{M}\beta_\ell\phi_\ell(y_{j-1}^{(r)}))\cdot e_i)^2$$

where e_i is the i^{th} canonical basis vector in $\mathbb{R}^d$.

We demonstrate the method for 1, 2 and 3 dimensional systems.

- For the 1-dimensional system, we use the ? oscillator:

$$\mathrm{d}X(t) = (\alpha X(t) + \beta X(t)^2 + \gamma)\,\mathrm{d}t + g\,\mathrm{d}W(t) \tag{13}$$

- For the 2-dimensional system, we use the undamped Duffing oscillator:

$$\begin{aligned}\mathrm{d}X_1(t) &= X_2(t)\mathrm{d}t + g_1\,\mathrm{d}W_1(t)\\ \mathrm{d}X_2(t) &= (-X_1(t) - X_1^3(t))\mathrm{d}t + g_2\,\mathrm{d}W_2(t)\end{aligned}$$

- For the 3-dimensional case, we consider 2 different form of equations. The first one is the damped Duffing oscillator, a general form of the damped oscillator considered in the 2-dimensional case:

$$\begin{aligned}\mathrm{d}X_1(t) &= X_2(t)\,\mathrm{d}t + g_1\,\mathrm{d}W_1(t)\\ \mathrm{d}X_2(t) &= (\alpha X_1(t) - \beta X_1(t) - \delta X_2(t) + \gamma\cos(X_3(t)))\,\mathrm{d}t + g_2\,\mathrm{d}W_2(t)\\ \mathrm{d}X_3(t) &= \omega\,\mathrm{d}t + g_3\,\mathrm{d}W_3(t)\end{aligned}$$

- Another example considered for the 3-dimensional case is the Lorenz oscillator:

$$\begin{aligned}\mathrm{d}X_1(t) &= \sigma(X_2(t) - X_1(t))\,\mathrm{d}t + g_1\,\mathrm{d}W_1(t)\\ \mathrm{d}X_2(t) &= (X_1(t)(\rho - X_3(t)))\mathrm{d}t + g_2\,\mathrm{d}W_2(t)\\ \mathrm{d}X_3(t) &= (X_1(t)X_2(t) - \beta X_3(t))\,\mathrm{d}t + g_3\,\mathrm{d}W_3(t)\end{aligned}$$

For simplicity, consider the example where the $X \in \mathbb{R}^2$ and the highest degree of the Hermite polynomial is three, including four Hermite polynomials:

$$\begin{aligned}f(x_1,x_2) &= \sum_{m=0}^{2}\sum_{i+j=0}^{i+j=m}\zeta_{i,j}\,\psi_{i,j}\\ &= \sum_{d=0}^{3}\sum_{i+j=0}^{i+j=3}\zeta_{i,j}H_i(x_1)H_j(x_2)\\ &= \sum_{i+j=0}\zeta_{i,j}H_i(x_1)H_j(x_2) + \sum_{i+j=1}\zeta_{i,j}H_i(x_1)H_j(x_2) + \sum_{i+j=2}\zeta_{i,j}H_i(x_1)H_j(x_2) + \sum_{i+j=3}\zeta_{i,j}H_i(x_1)H_j(x_2)\\ &= \zeta_{0,0}H_0(x_1)H_0(x_2) + \zeta_{0,1}H_0(x_1)H_1(x_2) + \zeta_{1,0}H_1(x_1)H_0(x_2) + \zeta_{0,2}H_0(x_1)H_2(x_2)\\ &+ \zeta_{2,0}H_2(x_1)H_0(x_2) + \zeta_{1,1}H_1(x_1)H_1(x_2) + \zeta_{0,3}H_0(x_1)H_3(x_2) + \zeta_{3,0}H_3(x_1)H_0(x_2)\\ &+ \zeta_{2,1}H_2(x_1)H_1(x_2) + \zeta_{1,2}H_1(x_1)H_2(x_2)\end{aligned}$$

References

[1] H. S. Bhat and R. W. M. A. Madushani. Nonparametric Adjoint-Based Inference for Stochastic Differential Equations. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 798–807, Oct. 2016. doi: 10.1109/DSAA.2016.69.

[2] R. N. Bhattacharya and E. C. Waymire. *Stochastic Processes with Applications*. SIAM, Aug. 2009. ISBN 978-0-89871-689-4. Google-Books-ID: 89dZjIid6XcC.

[3] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113 (15):3932–3937, Apr. 2016. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1517384113. URL `http://www.pnas.org/lookup/doi/10.1073/pnas.1517384113`.

[4] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Inferring Biological Networks by Sparse Identification of Nonlinear Dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2(1):52–63, June 2016. doi: 10.1109/TMBMC.2016.2633265.

[5] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor. Model selection for dynamical systems via sparse regression and information criteria. *Proc. R. Soc. A*, 473(2204): 20170009, Aug. 2017. ISSN 1364-5021, 1471-2946. doi: 10.1098/rspa.2017.0009. URL `http://rspa.royalsocietypublishing.org/content/473/2204/20170009`.

[6] F. v. d. Meulen, M. Schauer, and J. v. Waaij. Adaptive nonparametric drift estimation for diffusion processes using Faber–Schauder expansions. *Statistical Inference for Stochastic Processes*, pages 1–26, June 2017. ISSN 1387-0874, 1572-9311. doi: 10.1007/s11203-017-9163-7. URL `https://link.springer.com/article/10.1007/s11203-017-9163-7`.

[7] H.-G. Müller, F. Yao, and others. Empirical dynamics for longitudinal data. *The Annals of Statistics*, 38 (6):3458–3486, 2010. URL `http://projecteuclid.org/euclid.aos/1291126964`.

[8] J. Nicolau. NONPARAMETRIC ESTIMATION OF SECOND-ORDER STOCHASTIC DIFFERENTIAL EQUATIONS. *Econometric Theory*, 23(05):880, Oct. 2007. ISSN 0266-4666, 1469-4360. doi: 10.1017/S0266466607070375. URL `http://www.journals.cambridge.org/abstract_S0266466607070375`.

[9] O. Papaspiliopoulos and G. Roberts. Importance sampling techniques for estimation of diffusion models. *Statistical methods for stochastic differential equations*, 124:311–340, 2012.

[10] O. Papaspiliopoulos, G. O. Roberts, and O. Stramer. Data Augmentation for Diffusions. *Journal of Computational and Graphical Statistics*, 22(3):665–688, July 2013. ISSN 1061-8600, 1537-2715. doi: 10.1080/10618600.2013.783484. URL `http://www.tandfonline.com/doi/abs/10.1080/10618600.2013.783484`.

[11] G. O. Roberts and O. Stramer. On inference for partially observed nonlinear diffusion models using the Metropolis–Hastings algorithm. *Biometrika*, 88(3):603–621, Oct. 2001. ISSN 0006-3444. doi: 10.1093/biomet/88.3.603. URL `https://academic.oup.com/biomet/article/88/3/603/340094`.

[12] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, Apr. 2017. ISSN 2375-2548. doi: 10.1126/sciadv.1602614. URL `http://advances.sciencemag.org/content/3/4/e1602614`.

[13] H. Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 473 (2197):20160446, Jan. 2017. ISSN 1364-5021, 1471-2946. doi: 10.1098/rspa.2016.0446. URL `http://rspa.royalsocietypublishing.org/lookup/doi/10.1098/rspa.2016.0446`.

[14] H. Schaeffer, R. Caflisch, C. D. Hauck, and S. Osher. Sparse dynamics for partial differential equations. *Proceedings of the National Academy of Sciences*, 110(17):6634–6639, Apr. 2013. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1302752110. URL `http://www.pnas.org/cgi/doi/10.1073/pnas.1302752110`.

[15] H. Schaeffer, G. Tran, and R. Ward. Extracting Sparse High-Dimensional Dynamics from Limited Data. *arXiv:1707.08528 [math]*, July 2017. URL `http://arxiv.org/abs/1707.08528`. arXiv: 1707.08528.

[16] G. Tran and R. Ward. Exact Recovery of Chaotic Systems from Highly Corrupted Data. *Multiscale Modeling & Simulation*, 15(3):1108–1129, Jan. 2017. ISSN 1540-3459. doi: 10.1137/16M1086637. URL `https://epubs.siam.org/doi/abs/10.1137/16M1086637`.

[17] F. van der Meulen, M. Schauer, and H. van Zanten. Reversible jump MCMC for nonparametric drift estimation for diffusion processes. *Computational Statistics & Data Analysis*, 71:615–632, Mar. 2014. ISSN 0167-9473. doi: 10.1016/j.csda.2013.03.002. URL `http://www.sciencedirect.com/science/article/pii/S016794731300090X`.

[18] N. Verzelen, W. Tao, H.-G. Müller, and others. Inferring stochastic dynamics from functional data. *Biometrika*, 99(3):533–550, 2012. URL `http://nicolas.verzelen.free.fr/pdf/2012-Ver-Biometrika.pdf`.

[19] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Universitext. Springer-Verlag, Berlin Heidelberg, 6 edition, 2003. ISBN 978-3-540-04758-2. URL `//www.springer.com/us/book/9783540047582`.