Learning Stochastic Dynamical Systems via Bridge Sampling*

Harish S. Bhat[0000−0001−7631−1831] and Shagun Rawat

Applied Mathematics Unit, University of California, Merced, CA 95343
hbhat@ucmerced.edu

Abstract. We develop algorithms to automate discovery of stochastic dynamical system models from noisy, vector-valued time series. By discovery, we mean learning both a nonlinear drift vector field and a diagonal diffusion matrix for an Itô stochastic differential equation in $\mathbb{R}^d$. We parameterize the vector field using tensor products of Hermite polynomials, enabling the model to capture highly nonlinear and/or coupled dynamics. We solve the resulting estimation problem using expectation maximization (EM). This involves two steps. We augment the data via diffusion bridge sampling, with the goal of producing time series observed at a higher frequency than the original data. With this augmented data, the resulting expected log likelihood maximization problem reduces to a least squares problem. We provide an open-source implementation of this algorithm. Through experiments on systems with dimensions one through eight, we show that this EM approach enables accurate estimation for multiple time series with possibly irregular observation times. We study how the EM method performs as a function of the amount of data augmentation, as well as the volume and noisiness of the data.

Keywords: Stochastic differential equations, nonparametric estimation, diffusion bridges, expectation maximization

1 Introduction

Traditional mathematical modeling in the sciences and engineering often has as its goal the development of equations of motion that describe observed phenomena. Classically, these equations of motion usually took the form of deterministic systems of ordinary or partial differential equations (ODE or PDE, respectively). Especially in systems of contemporary interest in biology and finance where intrinsic noise must be modeled, we find stochastic differential equations (SDE) used instead of deterministic ones. Still, these models are often built from first principles, after which the model's predictions (obtained, for instance, by numerical simulation) are compared against observed data.

* H. S. Bhat was partially supported by NSF award DMS-1723272. Both authors acknowledge use of the MERCED computational cluster, funded by NSF award ACI-1429783.

Recent years have seen a surge of interest in using data to automate discovery of ODE, PDE, and SDE models. These machine learning approaches complement traditional modeling efforts, using available data to constrain the space of plausible models, and shortening the feedback loop linking model development to prediction and comparison to real observations. We posit two additional reasons to develop algorithms to learn SDE models. First, SDE models—including the models considered here—have the capacity to model highly nonlinear, coupled stochastic systems, including systems whose equilibria are non-Gaussian and/or multimodal. Second, SDE models often allow for interpretability. Especially if the terms on the right-hand side of the SDE are expressed in terms of commonly used functions (such as polynomials), we can obtain a qualitative understanding of how the system's variables influence, regulate, and/or mediate one other.

In this paper, we develop an algorithm to learn SDE models from high-dimensional time series. To our knowledge, this is the most general expectation maximization (EM) approach to learning an SDE with multidimensional drift vector field and diagonal diffusion matrix. Prior EM approaches were restricted to one-dimensional SDE [8], or used a Gaussian process approximation, linear drift approximation, and approximate maximization [25]. To develop our method, we use diffusion bridge sampling as in [13, 12], which focused on Bayesian nonparametric methods for SDE in $\mathbb{R}^1$. After augmenting the data using bridge sampling, we are left with a least-squares problem, generalizing the work of [6] from the ODE to the SDE context.

In the literature, variational Bayesian methods are the only other SDE learning methods that have been tested on high-dimensional problems [34]. These methods use approximations consisting of linear SDE with time-varying coefficients [1], kernel density estimates [2], or Gaussian processes [3]. In contrast, we parameterize the drift vector field using tensor products of Hermite polynomials; as mentioned above, the resulting SDE has much higher capacity than linear and/or Gaussian process models. Many other techniques explored in the statistical literature focus on scalar SDE [15, 14, 33, 4].

Differential equation discovery problems have attracted considerable recent interest. A variety of methods have been developed to learn ODE [6, 30, 7, 32, 28, 27, 18] as well as PDE [26, 20, 24, 19]. We do not describe these methods in detail here because, generally speaking, methods for learning deterministic models (such as ODE/PDE) to not readily generalize to the stochastic context considered in this paper. Note, however, that prior work on ODE/PDE learning has led to developments in model selection, which we do not address here. If needed, the method we propose can be combined with model selection procedures developed in the ODE context [10, 11].

2 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 Idt. 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 [5, 35]. 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.

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

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

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

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

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

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

Essentially, we parameterize f using tensor products of Hermite polynomials. 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. With respect to this inner product, the one-dimensional Hermite polynomials form an orthonormal basis of $L_w^2(\mathbb{R}) = \{f \ \langle f, f\rangle_w < \infty\}$. Consequently, by taking $\widetilde{M}$ sufficiently large, a vector field

whose j-th component is given by (5) can approximate any continuous vector field. *Hence the above model has the capacity to learn many SDE that occur in physics, including all Langevin equations driven by standard Brownian motions.*

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. In the derivation that follows, we will consider the data $(\mathbf{t}, \mathbf{x})$ to be one time series. Later, we indicate how our methods generalize naturally to multiple time series, i.e., repeated observations of the same system.

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 $h \ll \Delta t$ 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).

To augment or complete the data, we employ diffusion bridge sampling, using a Markov chain Monte Carlo (MCMC) method with origins in the work of [23, 17]. 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 [16], 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 [31].

Putting the above pieces together yields the following Metropolis algorithm (steps M1-3 below) 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 $\mathbf{z}^{(\ell)} = \{z_{i,j}^{(\ell)}\}_{j=1}^{F-1}$. Then:

M1 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 $\mathbf{z}^* = \{z_{i,j}^*\}_{j=1}^{F-1}$.

M2 Numerically approximate the integrals in (10) to compute the likelihood of the proposal. Specifically, we compute

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

We have discretized the stochastic $d\widehat{X}_s$ integral using Itô's definition, and we have discretized the ordinary ds integral using the trapezoidal rule.

M3 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 MCMC algorithm with a Brownian bridge path and use post-burn-in steps as the diffusion bridge samples we seek.

We now justify the intuition expressed above, that employing the diffusion bridge to augment the data on a fine scale will enable estimation. Let $\mathbf{z}^{(r)} = \{z_{i,j}^{(r)}\}_{j=1}^{F-1}$ be the r-th 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 $y_{1+iF}^{(r)} = x_i$ for $i = 0, 1, \ldots, L$, and that $y_{1+j+iF}^{(r)} = z_{i,j}$ for $j = 1, 2, \ldots, F-1$ and $i = 0, 1, \ldots, L-1$. With this notation, we can more easily express the EM algorithm.

Assume that we currently have access to $\boldsymbol{\theta}^{(k)}$, our estimate of the parameters after k iterations. If $k = 0$, we set $\boldsymbol{\theta}^{(0)}$ equal to an initial guess. Then we follow two steps:

E-step: For the expectation (E) step, we first generate an ensemble of R diffusion bridge sample paths. Interleaving as above, this yields R completed time series $\mathbf{y}^{(r)}$ for $r = 1, \ldots, R$. Define the Q function, or complete data expected log likelihood:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) = \mathbb{E}_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}^{(k)}} [\log p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})]. \tag{11}$$

In what follows, we will use an empirical average over diffusion bridge paths to approximate the expected value on the right-hand side of the Q function. 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:

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) &\approx \frac{1}{R} \sum_{r=1}^{R} \log p(\mathbf{y}^{(r)} \mid \boldsymbol{\theta}) \\ &= \frac{1}{R} \sum_{r=1}^{R} \sum_{n=1}^{N-1} \log p(y_{n+1}^{(r)} \mid y_n^{(r)}, \boldsymbol{\theta}) \\ &= -\frac{1}{R} \sum_{r=1}^{R} \sum_{n=1}^{N-1} \Bigg[\sum_{j=1}^{d} \frac{1}{2} \log(2\pi h_n \gamma_j^2) \\ &\quad + \frac{1}{2h_n} \left\| \Gamma^{-1} \Big(y_{n+1}^{(r)} - y_n^{(r)} - h_n \sum_{\ell=1}^{\widetilde{M}} \beta_{(\ell)} H_{(\ell)}\big(y_n^{(r)}\big) \Big) \right\|_2^2 \Bigg]. \end{aligned} \tag{12}$$

M-step: For the maximization (M) step, we have

$$\boldsymbol{\theta}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}).$$

Note that $y_j^{(r)} \in \mathbb{R}^d$—we denote the i-th component by $y_j^{(r),i}$. We find $\beta^{(k+1)}$ by solving $\mathcal{M}\beta = \rho$ where $\mathcal{M}$ is the $\widetilde{M} \times \widetilde{M}$ matrix

$$\mathcal{M}_{k,\ell} = \frac{1}{R} \sum_{r=1}^{R} \sum_{n=1}^{N-1} h_n H_{(k)}(y_n^{(r)}) H_{(\ell)}(y_n^{(r)}), \tag{13}$$

and ρ is the $\widetilde{M} \times d$ matrix

$$\rho_{k,i} = \frac{1}{R} \sum_{r=1}^{R} \sum_{n=1}^{N-1} H_{(k)}(y_n^{(r)})(y_{n+1}^{(r),i} - y_n^{(r),i}). \tag{14}$$

We find $\gamma^{(k+1)}$ by computing

$$\gamma_i^2 = \frac{1}{R(N-1)} \sum_{r=1}^{R} \sum_{n=1}^{N-1} h_n^{-1} (y_{n+1}^{(r),i} - y_n^{(r),i} - h_n \sum_{\ell=1}^{\widetilde{M}} \beta_{(\ell)}^i H_{(\ell)}(y_n^{(r)}))^2. \tag{15}$$

Here $\beta_{(\ell)}^i$ denotes the ℓ-th row and i-th column of the $\beta^{(k+1)}$ matrix. We then set $\boldsymbol{\theta}^{(k+1)} = (\beta^{(k+1)}, \gamma^{(k+1)})$.

We iterate EM steps until $\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\| / \|\boldsymbol{\theta}^{(k)}\| < \delta$ for some tolerance $\delta > 0$.

When the data consists of multiple time series $\{\mathbf{t}^{(i)}, \mathbf{x}^{(i)}\}_{i=1}^{S}$, everything scales accordingly. For instance, we create an ensemble of R diffusion bridge samples for each of the S time series. If we index the resulting completed time series appropriately, we simply replace R by RS in (13), (14), and (15) and keep everything else the same.

There are three sources of error in the above algorithm. The first relates to replacing the expectation by a sample average; the induced error should, by the law of large numbers, decrease as $R^{-1/2}$. The second stems from the approximate nature of the computed diffusion bridge samples—as indicated above, we use numerical integration to approximate the Girsanov likelihood. The third source of error is in using the Gaussian transition density to approximate the true transition density of the SDE. Both the second and third sources of error vanish in the $F \to \infty$ limit [9].

3 Experiments

We present a series of experiments with synthetic data. We have made available all source code required to reproduce our results and/or run further tests: https://github.com/hbhat4000/pathsamp/. Further details regarding simulations, experiments, and results are also available [21].

To generate this data, we start with a known stochastic dynamical system of the form (1). Using Euler-Maruyama time stepping starting from a randomly chosen initial condition, we march forward in time from $t = 0$ to a final time $t = 10$. In all examples, we step forward internally at a time step of $h = 0.0001$, but for the purposes of estimation, we only use data sampled every 0.1 units of time, discarding 99.9% of the simulated trajectory. We use a fine internal time step to reduce, to the extent possible, numerical error in the simulated data. We save the data on a coarse time scale to test the proposed EM algorithm.

To study how the EM method performs as a function of data augmentation, data volume, and noise strength, we perform four sets of experiments. In all experiments, we treat all noise strengths γ_j as known and estimate β only. When we run EM, we randomly generate the initial guess $\beta^{(0)} \sim \mathcal{N}(\mu = 0, \sigma^2 = 0.5)$. We set the EM tolerance parameter $\delta = 0.01$. The only regularization we include is to threshold β—values less than ν are set to zero. In the figures presented below, we refer to this value of ν as the threshold. Finally, in the MCMC diffusion bridge sampler, we use 10 burn-in steps and then create an ensemble of size $R = 100$.

To quantify the error between the estimated $\widetilde{\beta}$ and the true β, we use the Frobenius norm

$$\varepsilon = \sqrt{\sum_i \sum_\ell \left(\beta^i_{(\ell)} - \widetilde{\beta}^i_{(\ell)}\right)^2} \tag{16}$$

The $\widetilde{\beta}$ coefficients are the Hermite coefficients of the estimated drift vector field f. For each example system, we compute the true Hermite coefficients β by multiplying the true ordinary polynomial coefficients by a change-of-basis matrix that is easily computed.

We test the method using stochastic systems in dimensions $d = 1, 2, 3, 4, 8$. In 1D, we use

$$dX_t = (1 + X_t - X_t^2)dt + \gamma dW_t.$$

System \ F	1	2	3	4	5	6	7	8	9	10
1D	0.59	0.54	0.54	0.54	1.00	0.57	0.58	0.57	0.85	0.55
2D	0.65	0.57	0.58	0.57	0.57	0.57	0.57	0.62	0.56	0.57
3D	6.51	9.58	6.29	6.55	6.46	6.82	6.47	6.36	6.69	6.59
4D	24.08	24.34	23.94	23.98	24.93	25.65	23.99	23.17	25.64	24.54

Table 1: Results for average compute time (in seconds) per EM iteration for varying amount of data augmentation. As the Brownian bridge is created explicitly using the discretized version of (9), increasing the amount of data augmentation does significantly increase in the compute time. The time required to compute each EM iteration increases with the dimensionality of the system.

In 2D, we use a stochastic Duffing oscillator with no damping or driving:

$$dX_{0,t} = X_{1,t}dt + \gamma_0 dW_{0,t} \qquad dX_{1,t} = (-X_{0,t} - X_{0,t}^3)dt + \gamma_1 dW_{1,t}$$

For the 3D case, we consider the stochastic, damped, driven Duffing oscillator:

$$\begin{aligned} dX_{0,t} &= X_{1,t}dt + \gamma_0 dW_{0,t} \\ dX_{1,t} &= (X_{0,t} - X_{0,t}^3 - 0.3X_{1,t} + 0.5\cos(X_{2,t}))dt + \gamma_1 dW_{1,t} \\ dX_{2,t} &= 1.2dt + \gamma_2 dW_{2,t} \end{aligned}$$

Next, we consider linear, stochastic, coupled oscillator systems with $d = 2d'$. Assume we have a mass vector $m \in \mathbb{R}^{d'}$ and a spring constant vector $k \in \mathbb{R}^{d'+1}$. The network then consists of the following equations, for $j = 0, 1, 2, \ldots, d' - 1$, with the convention that $X_{i,t} \equiv 0$ if $i < 0$ or $i \geq d$:

$$\begin{aligned} dX_{2j,t} &= X_{2j+1,t} + \gamma_{2j} dW_{2j,t} \\ dX_{2j+1,t} &= [-k_j/m_j(X_{2j,t} - X_{2j-2,t}) - k_{j+1}/m_j(X_{2j,t} - X_{2j+2,t})]dt \\ &\quad + g_{2j+1} dW_{2j+1,t} \end{aligned}$$

We consider this system for both $d = 4$ and $d = 8$. In $d = 4$, we set $k = [1, 0.7, 0.6]$ and $m = [0.2, 0.3]$. In $d = 8$, we set $k = [1, 0.7, 0.6, 1.2, 0.9]$ and $m = [0.2, 0.3, 0.5, 1.1]$.

3.1 Experiment 1: Varying Data Augmentation

We start with $S = 10$ time series with $L + 1 = 51$ points each. Here we vary the number of interleaved diffusion bridge samples: $F = 1, \ldots, 10$. For $F = 1$, no diffusion bridge is created; the likelihood is computed by applying the Gaussian transition density directly to the observed data. The results, plotted in Figures 1 and 2, show that increased data augmentation dramatically improves the quality of estimated drifts for systems with $d = 1, 2, 3, 4, 8$. Though the Frobenius error

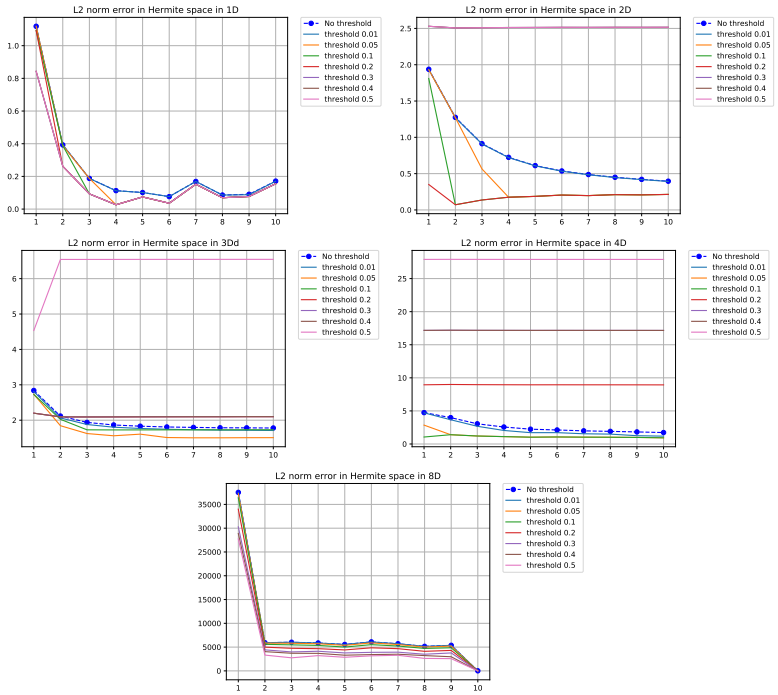

Fig. 1: As we increase the length F of the diffusion bridge interleaving observed data points, the quality of estimated drifts improves considerably. From left to right, top to bottom, we have plotted Frobenius errors (16) between true and estimated coefficients, for systems in $d = 1, 2, 3, 4, 8$.

for the 3D system exceeds 2.6, Figure 2 shows that EM's estimates are still accurate.

We have not plotted results for the scarce data regime where we have $S = 10$ time series with $L = 11$ points each. In this regime, data augmentation enables highly accurate estimation for the 2D and 3D systems. For the 1D system, the observations do not explore phase space properly, leading to poor estimation of the drift.

In Tables 1, 2, and 3, we report the average compute time (in seconds), the average MCMC acceptance rate, and the average number of iterations (for convergence), all as a function of F, the amount of data augmentation performed. Broadly speaking, none of these metrics show dependence on F. Instead, they depend primarily on d, the dimension of the problem under consideration.

The main point of EM, generally speaking, is to augment data. These experiments thus show that even with a basic diffusion bridge sampler, there is merit to the EM approach for estimating drift functions in diffusion processes. In ongoing/future work, we seek to explore using more sophisticated diffusion bridge

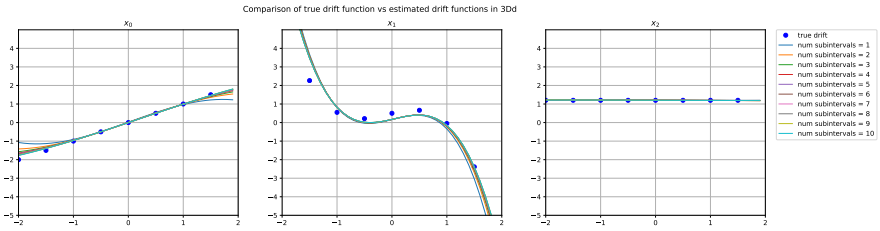

Fig. 2: Though Figure 1 shows the Frobenius norm error for the 3D system is greater than ≈ 2.6 at all noise levels, when plotted, the estimated drift functions lie close to the true drift function. The three components of the vector field are plotted as in the third row of Figure 3.

$\frac{F}{\text{System}}$	1	2	3	4	5	6	7	8	9	10
1D	100	75.04	67.08	61.78	61.23	58.71	55.22	53.57	52.58	49.52
2D	100	13.17	9.04	6.54	4.75	4.35	4.11	4.94	2.87	4.02
3D	100	6.07	3.20	2.82	2.74	2.54	2.48	2.27	2.51	2.41
4D	100	25.69	19.22	13.69	11.63	7.81	6.88	5.83	4.10	4.04

Table 2: Results for average acceptance rate for Metropolis-Hastings sampler for varying amount of data augmentation, F. For $F = 1$, no diffusion bridge has been created and thus the acceptance probability is 1. The algorithm in this case reduces to solving a least squares problem using only the observed time series. As we increase data augmentation, the acceptance probability decreases as it becomes more difficult to create a bridge between the observed values. The acceptance probability also decreases with an increase in the dimensionality and complexity of the system.

samplers, e.g., those that use the drift function to guide the proposal, rather than only incorporating the drift into the accept/reject ratio. Such approaches may help to reduce the d-dependence of the metrics we have plotted/tabulated.

3.2 Experiment 2: Varying Number of Time Series

Here we vary data volume by stepping the number S of time series from $S = 1$ to $S = 10$. Each time series has length $L + 1 = 101$. The results, as plotted in Figures 3 and 4, show that increasing S leads to improved estimates of β, as expected. As a rule of thumb, the results indicate that at least $S \geq 4$ time series are needed for accurate estimation.

3.3 Experiment 3: Varying Length of Time Series

Here we vary data volume by stepping the length $L + 1$ of the time series from $L + 1 = 11$ to $L + 1 = 101$, keeping the number of time series fixed at $S = 10$.

System \ F	1	2	3	4	5	6	7	8	9	10
1D	2	3	3	3	3	3	3	3	3	3
2D	2	8	5	7	6	8	8	4	9	6
3D	2	3	3	3	3	3	9	3	3	3
4D	2	2	2	2	2	2	2	2	2	2

Table 3: Results for number of EM iterations required to converge. We consider a threshold of 0.01, 0.05, 0.1 and 0.1 for the 1D, 2D, 3D and 4D systems respectively. Note that the number of iterations does not vary significantly as a function of the amount of data augmentation F.

Also note that in this experiment, observation times strictly between the initial and final times are chosen randomly. In Figure 5, we have plotted the estimated and true drifts for only the 3D system; in Figure 6, we have plotted the error (16) for all three systems. Comparing with Experiment 1, we see that randomization of the observation times improves estimation. That is, even with $L+1 = 11$ data points per time series, we obtain accurate estimates.

3.4 Experiment 4: Varying Noise Strength

Here we vary the noise strength γ, stepping from 0.5 to 0.0001 while keeping other parameters constant. Specifically, we take $S = 10$ time series each of length $L+1 = 101$. In Figure 7, we have plotted Frobenius errors for all three systems. Though the error in the estimated coefficients for the 3D system may seem large, the estimated and true drift functions are close—see Figure 8. Even when the algorithm does not recover ground truth parameter values, it yields a drift function that reproduces qualitative features of the ground truth drift.

4 Conclusion

We have developed an EM algorithm for estimation of drift functions and diffusion matrices for SDE. We have demonstrated the conditions under which the algorithm succeeds in estimating SDE. Specifically, our tests show that with enough data volume and data augmentation, the EM algorithm produces highly accurate results. Our tests also show that there is room for improvement, especially with regards to the basic Brownian bridge sampler incorporated here. In future work, we plan to study the effect of replacing the Brownian bridge sampler with a guided diffusion bridge sampler [29], especially with an eye towards increasing the MCMC acceptance rate for high-dimensional problems.

Here we have assumed we have direct access to discrete-time observations of the state $\mathbf{X}_t$ of the system. Such an assumption will be satisfied if we take as data low-dimensional projections of the solution process of a high-dimensional system; in this case, the method proposed in this paper can be used to derive

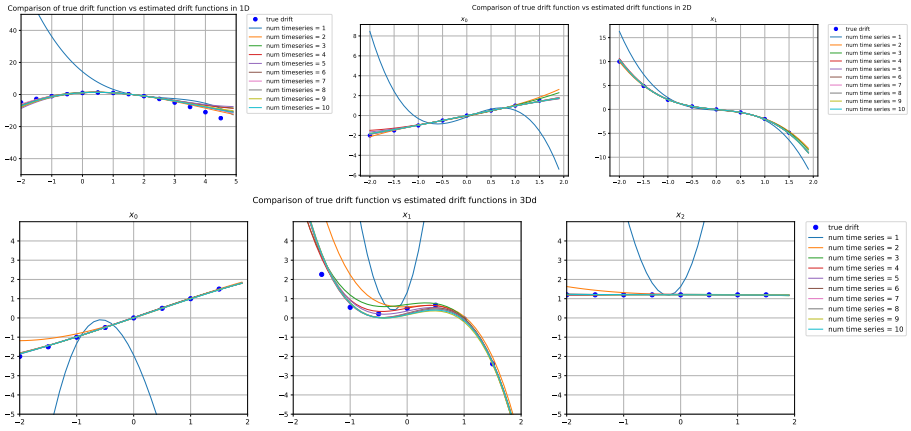

Fig. 3: As we increase the number S of time series used to learn the drift, the estimated drift more closely approximates the ground truth. From top to bottom, left to right, we have plotted estimated and true drifts for the 1D, 2D, and 3D systems. For the 1D and 2D systems, the true drifts depend on only one variable. For the $dX_{1,t}$ component of the 3D system, we have plotted the dependence of the drifts on X_0 only, keeping X_1 and X_2 fixed at 0.

SDE models for the evolution of the low-dimensional system. In future work, we also seek to further test our method on high-dimensional, nonlinear problems, problems with non-constant diffusion matrices, and real experimental data. In the latter case, we will explore coupling our EM method with a highly efficient batch filtering algorithm [22]. This will enable us to deal with observations of $\mathbf{Y}_t = \mathbf{X}_t + \varepsilon_t$, rather than observations of $\mathbf{X}_t$ itself.

References

1. Archambeau, C., Opper, M., Shen, Y., Cornford, D., Shawe-Taylor, J.S.: Variational inference for diffusion processes. In: Advances in Neural Information Processing Systems. pp. 17–24 (2008)
2. Batz, P., Ruttor, A., Opper, M.: Variational estimation of the drift for stochastic differential equations from the empirical density. Journal of Statistical Mechanics: Theory and Experiment **2016**(8), 083404 (2016)
3. Batz, P., Ruttor, A., Opper, M.: Approximate Bayes learning of stochastic differential equations. Physical Review E **98**, 022109 (2018)
4. Bhat, H.S., Madushani, R.W.M.A.: Nonparametric Adjoint-Based Inference for Stochastic Differential Equations. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA). pp. 798–807 (2016)
5. Bhattacharya, R.N., Waymire, E.C.: Stochastic Processes with Applications. SIAM (2009)
6. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the National Academy of Sciences **113**(15), 3932–3937 (2016)

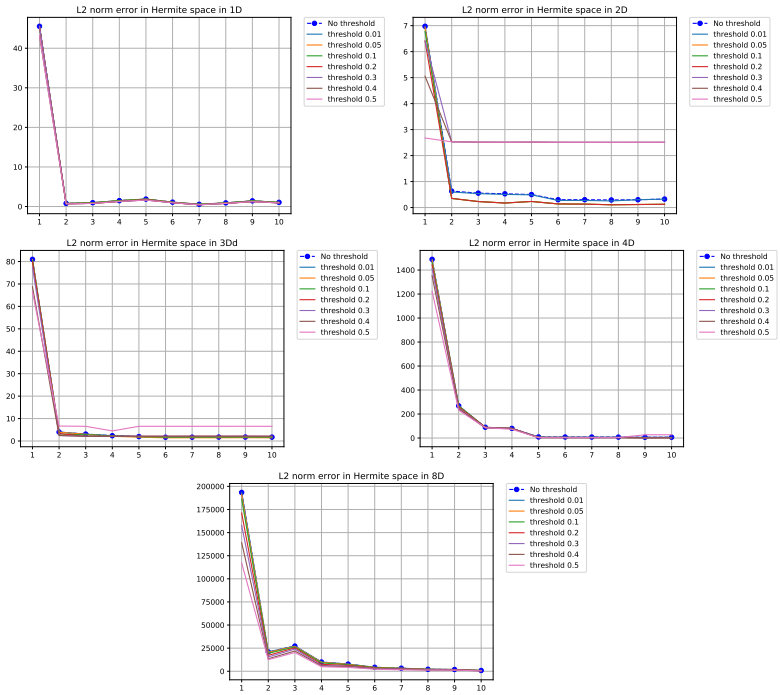

Fig. 4: As we increase the number S of time series used to learn the drift, the Frobenius norm error between estimated and true drifts—see (16)—decreases significantly. From left to right, top to bottom, we have plotted results for systems with $d = 1, 2, 3, 4, 8$.

7. Chen, S., Shojaie, A., Witten, D.M.: Network Reconstruction From High-Dimensional Ordinary Differential Equations. Journal of the American Statistical Association **112**(520), 1697–1707 (2017)
8. Ghahramani, Z., Roweis, S.T.: Learning nonlinear dynamical systems using an EM algorithm. Advances in Neural Information Processing Systems (NIPS) pp. 431–437 (1999)
9. Kloeden, P.E., Platen, E.: Numerical Solution of Stochastic Differential Equations. Springer Science & Business Media (2011)
10. Mangan, N.M., Brunton, S.L., Proctor, J.L., Kutz, J.N.: Inferring Biological Networks by Sparse Identification of Nonlinear Dynamics. IEEE Transactions on Molecular, Biological and Multi-Scale Communications **2**(1), 52–63 (2016)
11. Mangan, N.M., Kutz, J.N., Brunton, S.L., Proctor, J.L.: Model selection for dynamical systems via sparse regression and information criteria. Proc. R. Soc. A **473**(2204), 20170009 (2017)
12. van der Meulen, F., Schauer, M., van Waaij, J.: Adaptive nonparametric drift estimation for diffusion processes using Faber-Schauder expansions. Statistical Inference for Stochastic Processes pp. 1–26 (2017)
13. van der Meulen, F., Schauer, M., van Zanten, H.: Reversible jump MCMC for nonparametric drift estimation for diffusion processes. Computational Statistics &

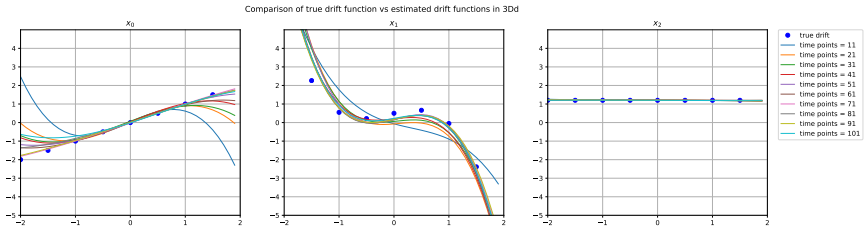

Fig. 5: We plot true and estimated drifts for the 3D system as a function of increasing time series length L. The three components of the vector field are plotted as in the third row of Figure 3. The results show that randomization of observation times compensates for a small value of L, enabling accurate estimation.

Data Analysis **71**, 615–632 (2014)
14. Müller, H.G., Yao, F., others: Empirical dynamics for longitudinal data. The Annals of Statistics **38**(6), 3458–3486 (2010)
15. Nicolau, J.: Nonparametric estimation of second-order stochastic differential equations. Econometric Theory **23**(05), 880 (2007)
16. Papaspiliopoulos, O., Roberts, G.O.: Importance sampling techniques for estimation of diffusion models. Statistical methods for stochastic differential equations **124**, 311–340 (2012)
17. Papaspiliopoulos, O., Roberts, G.O., Stramer, O.: Data Augmentation for Diffusions. Journal of Computational and Graphical Statistics **22**(3), 665–688 (2013)
18. Quade, M., Abel, M., Kutz, J.N., Brunton, S.L.: Sparse Identification of Nonlinear Dynamics for Rapid Model Recovery. Chaos **28**(6), 063116 (2018)
19. Raissi, M., Karniadakis, G.E.: Hidden physics models: Machine learning of nonlinear partial differential equations. Journal of Computational Physics **357**, 125–141 (2018)
20. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Machine learning of linear differential equations using Gaussian processes. Journal of Computational Physics **348**, 683–693 (2017)
21. Rawat, S.: Learning governing equations for stochastic dynamical systems. Ph.D. thesis, University of California, Merced (2018), advisor: Harish S. Bhat
22. Raziperchikolaei, R., Bhat, H.: A block coordinate descent proximal method for simultaneous filtering and parameter estimation. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 5380–5388. PMLR, Long Beach, California, USA (09–15 Jun 2019), http://proceedings.mlr.press/v97/raziperchikolaei19a.html
23. Roberts, G.O., Stramer, O.: On inference for partially observed nonlinear diffusion models using the Metropolis–Hastings algorithm. Biometrika **88**(3), 603–621 (2001)
24. Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N.: Data-driven discovery of partial differential equations. Science Advances **3**(4), e1602614 (2017)
25. Ruttor, A., Batz, P., Opper, M.: Approximate Gaussian process inference for the drift function in stochastic differential equations. In: Advances in Neural Information Processing Systems. pp. 2040–2048 (2013)

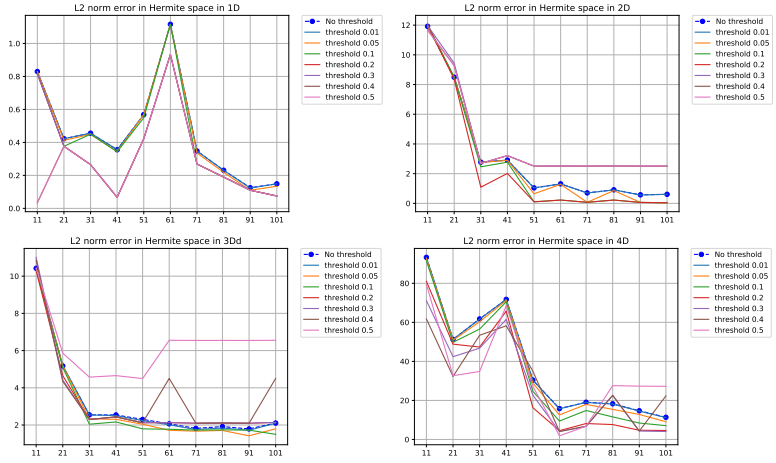

Fig. 6: As we increase the length L of each time series used for learning, the Frobenius norm error between estimated and true drifts—see (16)—decreases significantly. From left to right, we have plotted results for the 1D, 2D, 3D, and 4D systems.

26. Schaeffer, H., Caflisch, R., Hauck, C.D., Osher, S.: Sparse dynamics for partial differential equations. Proceedings of the National Academy of Sciences **110**(17), 6634–6639 (2013)
27. Schaeffer, H.: Learning partial differential equations via data discovery and sparse optimization. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science **473**(2197), 20160446 (2017)
28. Schaeffer, H., Tran, G., Ward, R.: Extracting Sparse High-Dimensional Dynamics from Limited Data. SIAM Journal on Applied Mathematics **78**(6), 3279–3295 (2018)
29. Schauer, M., van der Meulen, F., van Zanten, H.: Guided proposals for simulating multi-dimensional diffusion bridges. Bernoulli **23**(4A), 2917–2950 (2017). https://doi.org/10.3150/16-BEJ833
30. Schön, T.B., Svensson, A., Murray, L., Lindsten, F.: Probabilistic learning of nonlinear dynamical systems using sequential Monte Carlo. Mechanical Systems and Signal Processing **104**, 866–883 (2018)
31. Stuart, A.M.: Inverse problems: A Bayesian perspective. Acta Numerica **19**, 451–559 (2010)
32. Tran, G., Ward, R.: Exact Recovery of Chaotic Systems from Highly Corrupted Data. Multiscale Modeling & Simulation **15**(3), 1108–1129 (2017)
33. Verzelen, N., Tao, W., Müller, H.G., others: Inferring stochastic dynamics from functional data. Biometrika **99**(3), 533–550 (2012)
34. Vrettas, M.D., Opper, M., Cornford, D.: Variational mean-field algorithm for efficient inference in large systems of stochastic differential equations. Physical Review E **91**(1), 012148 (2015)
35. Øksendal, B.: Stochastic Differential Equations: An Introduction with Applications. Universitext, Springer-Verlag, Berlin Heidelberg, 6 edn. (2003)

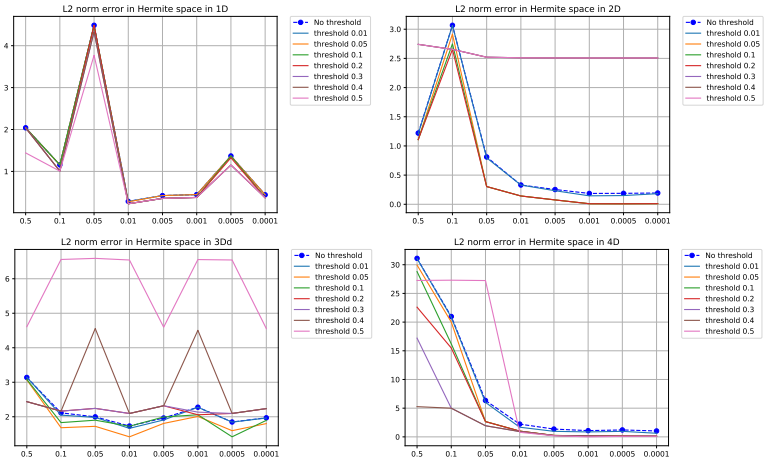

Fig. 7: Varying the strength of the noise in the simulated data alters the quality of estimated drift coefficients, quantified using the Frobenius error (16). We proceed from left to right. For the 1D and 2D systems, the maximum noise strength of 0.5 remains below the magnitude of the drift field coefficients. For these systems, as the noise strength decreases, the error drops close to zero. For the 3D system, the maximum noise strength of 0.5 is greater than or equal to two of the drift field coefficients, leading to apparently decreased performance—however, see Figure 8.

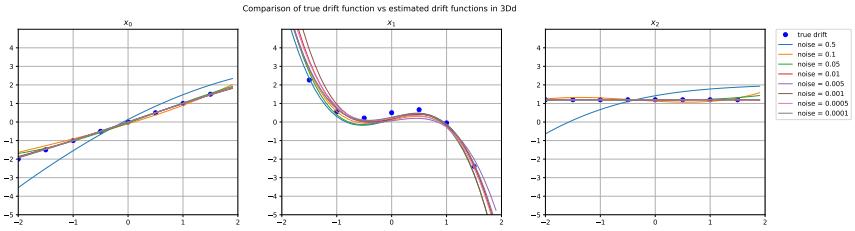

Fig. 8: Though Figure 7 shows a Frobenius norm error for the 3D system greater than ≈ 1.8 at all noise levels, when plotted, the estimated drift functions lie close to the true drift function. The three components of the vector field are plotted as in the third row of Figure 3.