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.
2. What is new and interesting about this work.

Points to cover:

- data specification
- Hermite polynomial and drift function representation
- Expectation and maximization formulas assuming data is filled in
- Filling data in with Brownian bridge
- MCMC iterations of brownian bridge using girsanov likelihood
- 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, ...
- Note: constant noise case, not inferring the gvec

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 [Bhattacharya and Waymire 2009, Oksendal 2007]. 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 $\Delta t = \frac{1}{L}\sum_{j=1}^{L}(t_j - t_{j-1})$ be the average 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. Note that $\widetilde{X}_{n+1}|\widetilde{X}_n = v$ has a $\mathcal{N}(v + f(v)h, h\Gamma^2)$ distribution. As h decreases, this Gaussian 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 [Roberts and Stramer, 2001; Papaspiliopoulos, Roberts, and Stramer, 2013]. Let us describe this method here. We suppose our current estimate of $\boldsymbol{\theta} = (\beta, \gamma)$ is given. Then the goal is to generate a sample path of (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}. 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

$$d\widehat{X}_t = \Gamma dW_t \tag{7}$$

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

$$\widehat{X}_t = \Gamma W_{t-t_i} + x_i - \frac{t-t_i}{t_{i+1}-t_i}(\Gamma W_{t_{i+1}-t_i} + x_i - x_{i+1}) \tag{8}$$

Let $\mathbf{z}^{(r)}$ denote the r^{th} diffusion bridge sample path:

$$z^{(r)} \sim z \mid x, \beta^{(k)} \tag{9}$$

The observed and sampled data can be interleaved together to create a time series (completed data)

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

of length $N = LF + 1$.

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 $\boldsymbol{\theta} = (\beta, \gamma)$.

1. Start with an initial guess for the parameters, $\boldsymbol{\theta}^{(0)}$.
2. For the expectation (or E) step,

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) = \mathbb{E}_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}^{(k)}}[\log p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})] \tag{10}$$

   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

$$\boldsymbol{\theta}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) \tag{11}$$

   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

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

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:

$$\mathrm{d}X(t) = (\alpha X(t) + \beta X(t)^2 + \gamma)\,\mathrm{d}t + g\,\mathrm{d}W(t) \tag{12}$$

- 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:

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

2 Expectation Maximization Steps

The data provided is in the form of a time series, $X \in \mathbb{R}^d$ at regular time points $t_l, 0 \leq l \leq L$. Note: EM step in the sampling writeup.

3 Brownian bridge sampling

When the inter-observation time of the observed data X is large, the expectation step becomes less accurate. To mitigate this problem, we can fill in the observed data with a Brownian bridge. We generate many samples of the $N-$dimensional Brownian bridge and accept-reject samples using the Metropolis-Hastings algorithm. The approximation of the likelihood is obtained using the Girsanov likelihood function.

3.1 Brownian bridge

The $\mathbb{R}^N$ dimensional Brownian bridge is defined by the integral:

$$I(t) = \int_0^t \frac{1-t}{1-T}\,\mathrm{d}W(t) \tag{13}$$

3.2 Metropolis Algorithm