2018 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2018, AALBORG, DENMARK

LEARNING STOCHASTIC DIFFERENTIAL EQUATIONS WITH GAUSSIAN PROCESSES WITHOUT GRADIENT MATCHING

Cagatay Yildiz[⋆] *Markus Heinonen*[⋆†] *Jukka Intosalmi*[⋆] *Henrik Mannerström*[⋆] *Harri Lähdesmäki*[⋆]

[⋆]Dept. of CS, Aalto University, Finland [†]Helsinki Inst. of Information Technology HIIT, Finland

ABSTRACT

We introduce a novel paradigm for learning non-parametric drift and diffusion functions for stochastic differential equation (SDE). The proposed model learns to simulate path distributions that match observations with non-uniform time increments and arbitrary sparseness, which is in contrast with gradient matching that does not optimize simulated responses. We formulate sensitivity equations for learning and demonstrate that our general stochastic distribution optimisation leads to robust and efficient learning of SDE systems.

Index Terms— Stochastic differential equations, Gaussian processes

1. INTRODUCTION

Dynamical systems modeling is a cornerstone of experimental sciences. Modelers attempt to capture the dynamical behavior of a stochastic system or a phenomenon in order to improve its understanding and make predictions about its future state. Stochastic differential equations (SDEs) are an effective formalism for modelling systems with underlying stochastic dynamics, with wide range of applications [1]. The key problem in SDE's is estimation of the underlying deterministic driving function, and the stochastic diffusion component.

We consider the dynamics of a multivariate system governed by Markov process $\mathbf{x}_t$ described by an SDE

time-homogeneous Langevin.

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t)dt + \sigma(\mathbf{x}_t)dW_t \tag{1}$$

where $\mathbf{x}_t \in \mathbb{R}^D$ is the state vector of a D-dimensional dynamical system at continuous time $t \in \mathbb{R}$, $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^D$ is a deterministic state evolution, $\sigma(\mathbf{x}) \in \mathbb{R}$ is a scalar magnitude of the stochastic multivariate Wiener process $W_t \in \mathbb{R}^D$. The Wiener process has zero initial state $W_0 = \mathbf{0}$, and the independent increments $W_{t+s} - W_t \sim \mathcal{N}(\mathbf{0}, sI)$ follow a Gaussian with standard deviation $\sqrt{s}$.

The SDE system (1) transforms states $\mathbf{x}_t$ forward in continuous time by the deterministic *drift* component $\mathbf{f}$, while the σ is the magnitude of the random Brownian *diffusion* W_t that scatters the state $\mathbf{x}_t$ with random fluctuations. The state solutions of SDE are given by the Itô integral [2]

$$\mathbf{x}_t = \mathbf{x}_0 + \int_0^t \mathbf{f}(\mathbf{x}_\tau)d\tau + \int_0^t \sigma(\mathbf{x}_\tau)dW_\tau, \tag{2}$$

where we integrate the system state from an initial state $\mathbf{x}_0$ for time t forward, and where τ is an auxiliary time variable. The only non-deterministic part of the solution (2) is the Brownian motion W_τ, whose random realisations generate path realisations $\mathbf{x}_{0\ldots t}$ that induce state distributions $p(\mathbf{x}|t; \mathbf{f}, \sigma)$ at time t given the drift $\mathbf{f}$ and diffusion σ. SDEs produce continuous, but non-smooth trajectories $\mathbf{x}_{0\ldots t}$ over time due to the non-differentiable Brownian motion.

We assume that both $\mathbf{f}(\cdot)$ and $\sigma(\cdot)$ are *completely unknown* and we only observe one or several multivariate time series $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^T \in \mathbb{R}^{N\times D}$ obtained from noisy observations at observation times $T = (t_1, \ldots, t_N) \in \mathbb{R}^N$,

N.b., noisy observation on top of noisy dynamics.

See postit to bottom left of Eq. 15 - this need NOT be "observational noise",

$$\mathbf{y}_t = \mathbf{x}_t + \varepsilon_t, \tag{3}$$

where $\varepsilon_t \sim \mathcal{N}(\mathbf{0}, \Omega)$ follows a stationary time-invariant zero-mean multivariate Gaussian distribution with diagonal noise variances $\Omega = \text{diag}(\omega_1^2, \ldots, \omega_D^2)$; and the latent states $\mathbf{x}_t \sim p(\mathbf{x}|t; \mathbf{f}, \sigma)$ follow the state distribution. The goal of SDE modelling is to learn the drift $\mathbf{f}$ and diffusion σ functions such that the process $\mathbf{x}_t$ matches data $\mathbf{y}_t$.

There is considerable amount of literature on inferring SDEs that have a pre-defined parametric drift or diffusion functions for specific applications [1]. There has also been interest on estimating non-parametric SDE drift and diffusion functions from data using the general Bayesian formalism [3, 4]. With linear drift approximations the state distribution turns out to be a Gaussian, which can be solved with variational smoothing algorithm [5] or by variational mean field approximation [6]. Non-linear drifts and diffusions are predominantly modelled with Gaussian processes [3, 7, 4], which are a family of Bayesian kernel methods [8]. For such models the state distributions are intractable, and hence these methods resort to using a family of gradient matching approximations [9, 10, 11], where the drift is estimated to match the empirical gradients of data, $\mathbf{f}(\mathbf{y}_i) \approx (\mathbf{y}_{i+1} - \mathbf{y}_i)\Delta t_i$, and the diffusion relates to the residual of the approximation [3, 7, 4]. The gradient matching is only applicable to dense observa-

So a GP "posterior" based on ficticious data of pairs of drift vectors and diffusion scalars, one pair at each of the inducing locations. In other words, the ficticious data are actually parameters of a model. Choose those parameters to maximize the resultant posterior of the empirical time-series, i.e., of the real data, where the likelihood is given by Eq. 4, and prior is given on the next page.

tions over time, while additional linearisation [3, 7] is necessary to model sparse observations. Non-parametric estimation of diffusion only applies to dense data [7, 4].

In this paper we propose to infer non-parametric drift and diffusion functions with Gaussian processes for arbitrary sparse or dense data. We learn the underlying system to induce state distributions with high expected likelihood,

N IID time-series, of length N each.

$$p(Y|\mathbf{f},\sigma,\Omega) = \prod_{i=1}^{N} \mathbb{E}_{p(\mathbf{x}|t_i;\mathbf{f},\sigma)}[\mathcal{N}(\mathbf{y}_i|\mathbf{x},\Omega)]. \tag{4}$$

Not sure how they distinguish the prob. of \vec{x}'s that are simply translated versions of one another in this likelihood model.

The expected likelihood is generally intractable. In contrast to earlier works we do not use gradient matching or other approximative models, but instead we directly tackle and optimize the SDE system against the true likelihood (4) by performing full forward simulation. We propose an unbiased stochastic Monte Carlo approximation for the likelihood, for which we derive efficient, tractable gradients. Our approach places no restrictions on the spacing or sparsity of the observations. Our model is denoted as NPSDE, and the implementation is publicly available in `http://www.github.com/cagatayyildiz/npde`.

2. INDUCING GAUSSIAN PROCESS SDE MODEL

In this section we model both the drift $\mathbf{f}(\mathbf{x})$ and diffusion $\sigma(\mathbf{x})$ as Gaussian processes with a general inducing point parameterisation. The drift function defines a *vector field* $\mathbf{f} : \mathbb{R}^D \to \mathbb{R}^D$, that is, an assignment of a D-dimensional gradient vector $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^D$ to every D-dimensional state $\mathbf{x} \in \mathbb{R}^D$. We assume that drift does not depend on time. The diffusion function $\sigma(\mathbf{x}) \in \mathbb{R}$ is a standard scalar function. We model both functions as Gaussian processes (GP), which are flexible Bayesian non-linear and non-parametric models [8].

2.1. Drift Gaussian process

The inducing point parameterisation for the drift GP was originally proposed in the context of ordinary differential equation systems [12], which we review here. We assume a zero-mean vector-valued GP prior on the drift function

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, K_\mathbf{f}(\mathbf{x},\mathbf{x}')), \tag{5}$$

which defines *a priori* distribution over drift values $\mathbf{f}(\mathbf{x})$ whose mean and covariance are

$$\mathbb{E}[\mathbf{f}(\mathbf{x})] = \mathbf{0} \tag{6}$$

$$\mathrm{cov}[\mathbf{f}(\mathbf{x}),\mathbf{f}(\mathbf{x}')] = K_\mathbf{f}(\mathbf{x},\mathbf{x}') \quad \in \mathbb{R}^{D\times D}, \tag{7}$$

where the kernel $K_\mathbf{f}(\mathbf{x},\mathbf{x}')$ is matrix-valued [13]. A GP prior defines that for any collection of states $X = (\mathbf{x}_1,\ldots,\mathbf{x}_N)^T \in \mathbb{R}^{N\times D}$, the drift values $F = (\mathbf{f}(\mathbf{x}_1),\ldots,\mathbf{f}(\mathbf{x}_N))^T \in \mathbb{R}^{N\times D}$ follow a matrix-valued normal [13],

$$p(F) = \mathcal{N}(\mathrm{vec}\, F|\mathbf{0}, \mathbf{K}_\mathbf{f}(X,X)), \tag{8}$$

where $\mathbf{K}_\mathbf{f}(X,X) = (K_\mathbf{f}(\mathbf{x}_i,\mathbf{x}_j))_{i,j=1}^N \in \mathbb{R}^{ND\times ND}$ is a block matrix of matrix-valued kernels $K_\mathbf{f}(\mathbf{x}_i,\mathbf{x}_j)$. The key property of Gaussian processes is that they encode functions where similar states $\mathbf{x},\mathbf{x}'$ induce similar drifts $\mathbf{f}(\mathbf{x}),\mathbf{f}(\mathbf{x}')$, and where the state similarity is defined by the kernels $K_\mathbf{f}(\mathbf{x},\mathbf{x}')$. Several families of rich matrix-valued kernels exist [14, 13, 16]. In this work we opt for the family of decomposable kernels $K(\mathbf{x},\mathbf{x}') = k(\mathbf{x},\mathbf{x}')\cdot A$, where $k(\mathbf{x},\mathbf{x}')$ is a Gaussian base kernel

$$k(\mathbf{x},\mathbf{x}') = \sigma_\mathbf{f}^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{(x_d - x_d')^2}{\ell_{\mathbf{f}d}^2}\right) \tag{9}$$

with drift variance $\sigma_\mathbf{f}^2$, and dimension-specific lengthscales $\ell_{\mathbf{f}1},\ldots,\ell_{\mathbf{f}D}$ that determine the smoothness of drift field, and $A \in \mathbb{R}^{D\times D}$ is a PSD dependency matrix between dimensions. In practise global dependency structures are often unavailable, and the diagonal structure $A = I_D$ is then chosen.

PSD = Positive semi-definite

In standard GP regression we would obtain posterior of the drift by conditioning the GP prior with data [8]. In SDE models the conditional $\mathbf{f}(\mathbf{x})|Y$ is intractable due to the integral mapping (2) between observations $\mathbf{y}_i$ and drifts $\mathbf{f}(\mathbf{x})$. Instead, we augment the Gaussian process with a set of M *inducing vectors* $\mathbf{u}_\mathbf{f} \in \mathbb{R}^D$ at locations $\mathbf{z} \in \mathbb{R}^D$, such that $\mathbf{f}(\mathbf{z}) = \mathbf{u}_\mathbf{f}$ [17]. We interpolate drift from inducing points as

K(., .) defined just before Eq. 9

$$\mathbf{f}(\mathbf{x}) \triangleq \mathbf{K}(\mathbf{x},Z)\mathbf{K}(Z,Z)^{-1}\mathbf{u}_\mathbf{f}, \tag{10}$$

which supports the drift with *inducing locations* $Z = (\mathbf{z}_1,\ldots,\mathbf{z}_M)$ and *inducing vectors* $U_\mathbf{f} = (\mathbf{u}_{\mathbf{f}1},\ldots,\mathbf{u}_{\mathbf{f}M})$, and where $\mathbf{u}_\mathbf{f} = \mathrm{vec}\, U_\mathbf{f}$. This corresponds to a vector-valued kernel function [13], or to a multi-task Gaussian process posterior mean [8]. Due to universality of the Gaussian kernel [18], we can represent arbitrary drifts with sufficient inducing points.

2.2. Diffusion Gaussian process

We represent diffusion $\sigma(\mathbf{x})$ as another inducing point GP, similarly to drift. Diffusion is a scalar function that uses a scalar kernel. The diffusion has a zero-mean GP prior

$$\sigma(\mathbf{x}) \sim \mathcal{GP}(0, k_\sigma(\mathbf{x},\mathbf{x}')) \tag{11}$$

that defines covariance $\mathrm{cov}[\sigma(\mathbf{x}),\sigma(\mathbf{x}')] = k_\sigma(\mathbf{x},\mathbf{x}') \in \mathbb{R}$ with a Gaussian kernel of form (9), but with diffusion variance σ_σ^2 and lengthscales $\{\ell_{\sigma d}\}$. Diffusion values $\boldsymbol{\sigma} = (\sigma(\mathbf{x}_1),\ldots,\sigma(\mathbf{x}_N))^T \in \mathbb{R}^N$ at states X then follow a prior

$$p(\boldsymbol{\sigma}) = \mathcal{N}(\boldsymbol{\sigma}|\mathbf{0}, K_\sigma(X,X)), \tag{12}$$

where $K_\sigma(X,X) = (k_\sigma(\mathbf{x}_i,\mathbf{x}_j))_{i,j=1}^N \in \mathbb{R}^{N\times N}$. We interpolate the diffusion from M inducing locations Z with inducing values $\mathbf{u}_\sigma = (u_{\sigma1},\ldots,u_{\sigma M})^T \in \mathbb{R}^M$,

$$\sigma(\mathbf{x}) \triangleq K_\sigma(\mathbf{x},Z)K_\sigma(Z,Z)^{-1}\mathbf{u}_\sigma, \tag{13}$$

Drift and diffusion naturally share their inducing locations Z.

N.b., he gets his parameters based on samples of the *entire* trajectories, starting from the initial points of each time-series, rather than by creating sample trajectories starting from each data point in the time series.

N.b., in their experiments, they assumed the distribution at t = 0 is a delta function about x = 0.

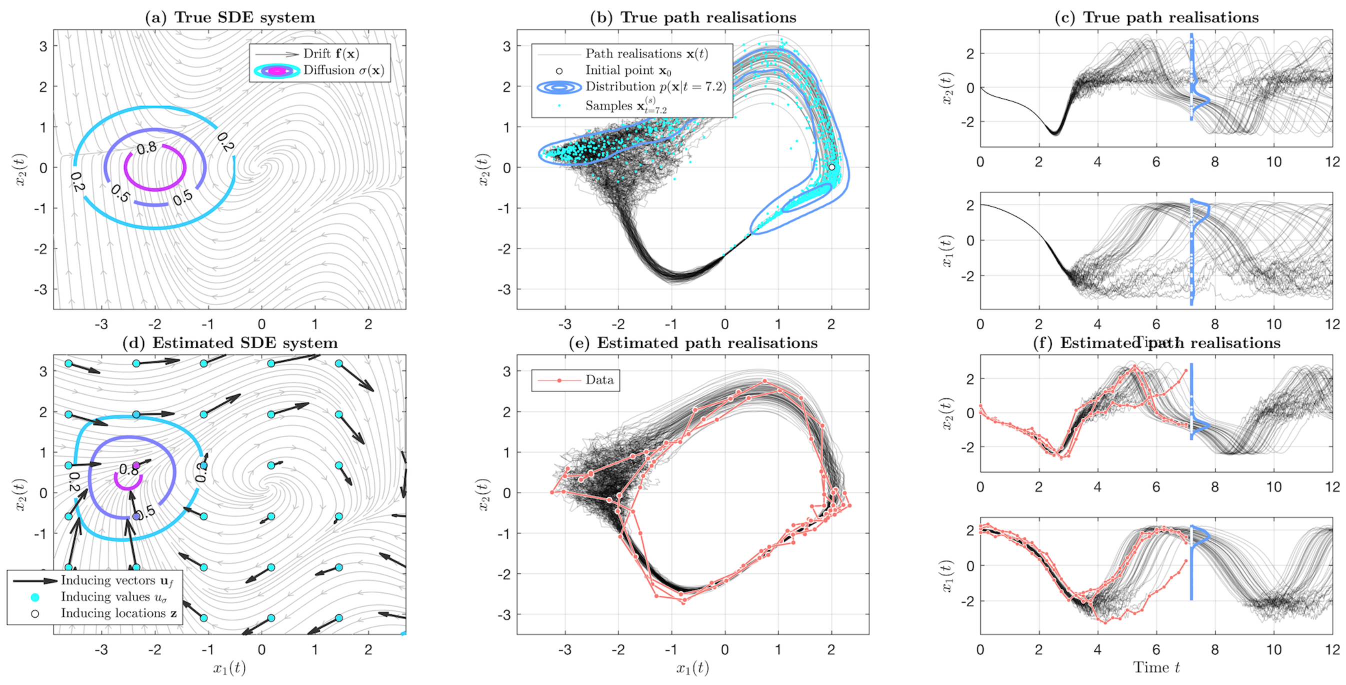

Fig. 1. **(a)** A Van der Pol oscillator with local diffusion and drift, **(b-c)** path samples and distribution, **(d)** the estimated GP system, **(e)** three noisy input trajectories for training, **(e-f)** the estimated path samples that match the true samples.

2.3. Stochastic Monte Carlo inference

The inducing SDE model is determined via the inducing locations Z, the inducing values $\mathbf{u_f}$ and $\mathbf{u}_\sigma$, the observation noise variance σ_n^2, and the kernel parameters $\sigma_\sigma, \{\ell_{\sigma d}\}$ and $\sigma_\mathbf{f}, \{\ell_{\mathbf{f}d}\}$ of the drift and diffusion kernels. The posterior of the model combines the likelihood $p(Y|\mathbf{f}, \sigma, \Omega)$ of (4) and the independent priors $p(\mathbf{u_f})$ and $p(\mathbf{u}_\sigma)$ using Bayes' theorem as

$$p(\mathbf{u_f}, \mathbf{u}_\sigma|Y) \propto p(\mathbf{u_f}, \mathbf{u}_\sigma) p(Y|\mathbf{u_f}, \mathbf{u}_\sigma) \quad (14)$$

$$= p(\mathbf{u_f}) p(\mathbf{u}_\sigma) \prod_{i=1}^{N} \mathbb{E}_{p(\mathbf{x}|t_i;\mathbf{f},\sigma)}[\mathcal{N}(\mathbf{y}_i|\mathbf{x}, \Omega)]$$

$$\approx \mathcal{N}(\mathbf{u_f}|\mathbf{0}, \mathbf{K_f}(Z,Z)) \mathcal{N}(\mathbf{u}_\sigma|\mathbf{0}, K_\sigma(Z,Z)) \quad (15)$$

$$\times \prod_{i=1}^{N} \frac{1}{N_s} \sum_{s=1}^{N_s} \mathcal{N}(\mathbf{y}_i|\mathbf{x}_i^{(s)}, \Omega), \quad \mathbf{x}^{(s)} \sim p(\mathbf{x}_{0\ldots t}|\mathbf{u_f}, \mathbf{u}_\sigma, Z)$$

K_f defined on top right on p. 2

where the time-dependent state distribution $p(\mathbf{x}|t; \mathbf{f}, \sigma) \equiv p(\mathbf{x}|t; \mathbf{u_f}, \mathbf{u}_\sigma, Z)$ now depends on the inducing parameters. We propose to approximate the true expected likelihood with an unbiased stochastic Monte Carlo averaging, since we can draw path samples $\mathbf{x}_t^{(s)}$ from the state distribution $p(\mathbf{x}_{0\ldots t}|\mathbf{u_f}, \mathbf{u}_\sigma, Z)$ by sampling the Brownian motion path $W_t^{(s)}$. The stochastic likelihood estimate with N_s samples turns out to be a kernel density estimator with Gaussian bases.

We draw the sample paths using Euler-Maruyama(EM) method for approximating the solution of an SDE (2) [2]:

$$\mathbf{x}_{i+1}^{(s)} = \mathbf{x}_i^{(s)} + \mathbf{f}(\mathbf{x}_i^{(s)})\Delta t + \sigma(\mathbf{x}_i^{(s)})\Delta W_i^{(s)}, \quad (16)$$

where we discretise time into N_T subintervals $t_0, t_1, \ldots, t_{N_T}$ of width $\Delta t = t_{N_T}/N_T$, and sample the Wiener coefficients as $\Delta W_i^{(s)} \sim \mathcal{N}(\mathbf{0}, \Delta t \cdot I)$ with standard deviation $\sqrt{\Delta t}$. We set $\mathbf{x}_0^{(s)}$ to the initial observation and use (16) to compute state path $\mathbf{x}^{(s)} \equiv (\mathbf{x}_0^{(s)}, \mathbf{x}_1^{(s)}, \ldots, \mathbf{x}_{T_N}^{(s)})$. The number of time steps $N_T > N$ is often higher than the number of observed time-points to achieve sufficient path resolution.

We find a maximum *a posteriori* (MAP) estimates of $\mathbf{u_f}, \mathbf{u}_\sigma, \Omega$ by gradient ascent, while choosing lengthscales $\boldsymbol{\ell}_\mathbf{f}, \boldsymbol{\ell}_\sigma$ from a grid, keeping the inducing locations Z fixed on a dense grid (See Figure 1(d)) and setting $\sigma_\mathbf{f} = \sigma_\sigma = 1$. In practise in $2D$ or $3D$ systems placing inducing locations on a grid is a robust choice, noting that they can be also optimised with increased computational complexity [19].

2.4. Computing stochastic gradients

The gradient of the expectation of the log-likelihood (15) is

$$\frac{d}{d\mathbf{u}} \sum_{i=1}^{N} \log \frac{1}{N_s} \sum_{s=1}^{N_s} \mathcal{N}(\mathbf{y}_i|\mathbf{x}_i^{(s)}, \Omega) \quad (17)$$

$$= \sum_{i=1}^{N} \frac{\sum_{s=1}^{N_s} \frac{\partial \mathcal{N}(\mathbf{y}_i|\mathbf{x}_i^{(s)}, \Omega)}{\partial \mathbf{x}} \frac{d\mathbf{x}_i^{(s)}}{d\mathbf{u}}}{\sum_{s=1}^{N_s} \mathcal{N}(\mathbf{y}_i|\mathbf{x}_i^{(s)}, \Omega)}, \quad (18)$$

Based on the code,, it appears that he trains the value of Omega while training the other parameters

Combining Eq. 17 with Eq. 16 ... it seems that they're assuming regularly sampled data, with no missing values - talking with Yildiz though, it seems they allow irregularly sampled data.

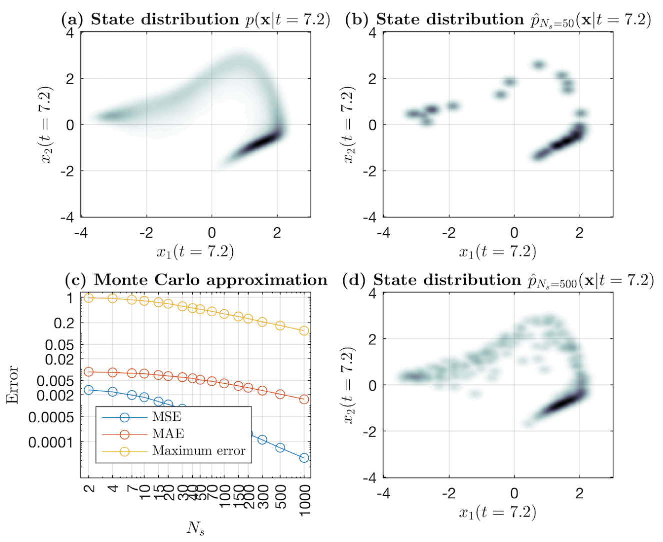

Fig. 2. **(a)** True state distribution from the model in Figure 1(a-b), **(b,d)** state distribution approximations with different number of path samples, **(c)** approximation errors.

where the sample paths $\mathbf{x}_t^{(s)}$ are from equation (16). The last term $\frac{d\mathbf{x}_i^{(s)}}{d\mathbf{u}}$ is the cumulative derivative of the state $\mathbf{x}_i^{(s)}$ of sample s at time t_i against the parameters $\mathbf{u} \triangleq (\mathbf{u_f}, \mathbf{u}_\sigma)$. The gradients of the piecewise Euler-Maruyama paths $\mathbf{x}_i^{(s)}$ are:

$$\begin{aligned} \frac{d\mathbf{x}_{i+1}^{(s)}}{d\mathbf{u}} &= \frac{d\mathbf{x}_i^{(s)}}{d\mathbf{u}} + \frac{d\mathbf{f}(\mathbf{x}_i^{(s)})}{d\mathbf{u}}\Delta t + \frac{d\sigma(\mathbf{x}_i^{(s)})}{d\mathbf{u}}\Delta W_i^{(s)} \\ &= \frac{d\mathbf{x}_i^{(s)}}{d\mathbf{u}} + \left(\frac{\partial \mathbf{f}(\mathbf{x}_i^{(s)})}{\partial \mathbf{x}}\frac{d\mathbf{x}_i^{(s)}}{d\mathbf{u_f}} + \frac{\partial \mathbf{f}(\mathbf{x}_i^{(s)})}{\partial \mathbf{u_f}}\right)\Delta t \quad (19) \\ &\quad + \left(\frac{\partial \sigma(\mathbf{x}_i^{(s)})}{\partial \mathbf{x}}\frac{d\mathbf{x}_i^{(s)}}{d\mathbf{u}_\sigma} + \frac{\partial \sigma(\mathbf{x}_i^{(s)})}{\partial \mathbf{u}_\sigma}\right)\Delta W_i^{(s)}. \end{aligned}$$

The derivatives $\frac{d\mathbf{x}_i^{(s)}}{d\mathbf{u}}$ are constructed iteratively over time starting from $\frac{d\mathbf{x}_0^{(s)}}{d\mathbf{u}} = \mathbf{0}$ with a fixed initial state $\mathbf{x}_0^{(s)}$, and the four partial derivatives are gradients of the kernel functions (10) and (13) with respect to $\mathbf{u_f}$ and $\mathbf{u}_\sigma$, respectively:

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial \mathbf{K_f}(\mathbf{x}, Z)}{\partial \mathbf{x}}\mathbf{K_f}(Z, Z)^{-1}\mathbf{u_f} \quad (20)$$

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{u_f}} = \mathbf{K_f}(\mathbf{x}, Z)\mathbf{K_f}(Z, Z)^{-1} \quad (21)$$

$$\frac{\partial \sigma(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial K_\sigma(\mathbf{x}, Z)}{\partial \mathbf{x}}K_\sigma(Z, Z)^{-1}\mathbf{u}_\sigma \quad (22)$$

$$\frac{\partial \sigma(\mathbf{x})}{\partial \mathbf{u}_\sigma} = K_\sigma(\mathbf{x}, Z)K_\sigma(Z, Z)^{-1}. \quad (23)$$

These gradients are related to the sensitivity equations [20, 21] derived for non-parametric ODEs previously [12]. The gradients (19) can be computed in practise together with the sample paths (16) during the Euler-Maruyama iteration. The computation of the gradients has the same computational complexity as the numerical simulator. The iterative gradients are superior to finite difference approximation since we have exact formulation of the gradients, albeit of the approximal Euler-Maruyama paths, which can be solved to arbitrary numerical accuracy by tuning Δt discretisation.

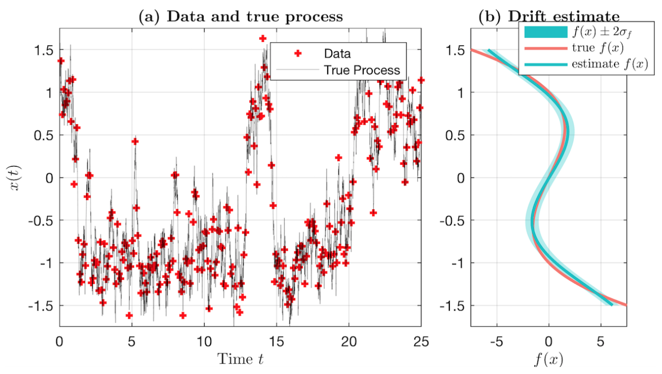

Fig. 3. **(a)** Double well system, **(b)** estimated drift.

3. EXPERIMENTS

In order to illustrate the performance of our model, we conduct several experiments on synthetic data as well as real-world data sets. In all experiments inducing vectors are initialized by gradient matching, and then fully optimised. We use EM method to simulate the state distributions and compute stochastic gradients over $N_s = 50$ samples. We use L-BFGS algorithm to compute the MAP estimates.

3.1. Double well

We first consider the double well system where the drift is given by $f(x) = 4(x - x^3)$ and with constant diffusion $\sigma(x) = 1.5$. We generate 6 random noisy input trajectories, each with 250 observed data points. True dynamics and the observed data points are illustrated in Figure 3a. We fit our npSDE model with $M = 15$ inducing points located uniformly within $[-5, 5]$. We accurately approximate the true drift (the right plot on Figure 3), and learn a diffusion estimate of 1.39.

3.2. Simple oscillating dynamics

Next, we investigate how the quality of fit changes by the amount of data used for training. We consider the $2D$ synthetic system in [7], whose drift equations are given by $\mathbf{f}(\mathbf{x})_1 = x_1(1-x_1^2-x_2^2)-x_2$ and $\mathbf{f}(\mathbf{x})_2 = x_2(1-x_1^2-x_2^2)+x_1$, and the diffusion is $\sigma(\mathbf{x}) = 2\mathcal{N}(\mathbf{x}|[-1, -1], 0.5I) + 0.3$. The state-dependent diffusion acts an hotspot of increased

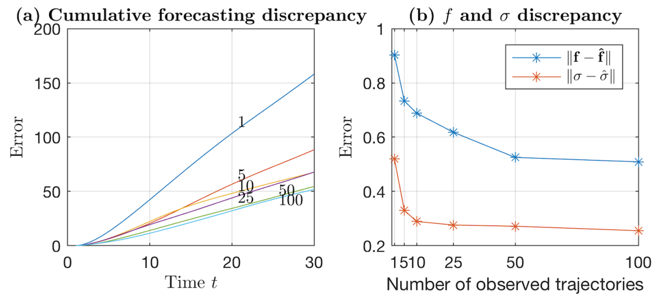

Fig. 4. **(a)** Cumulative discrepancy in the state distributions over time and over number of observed trajectories in synthetic $2D$ model, **(b)** drift and diffusion estimation errors.

path scatter, and provides interesting and challenging dynamics to infer. We generate six data batches from the true dynamics using EM method with step size $\Delta t = 0.005$, and observe every 100'th state corrupted by a Gaussian noise with variance $\sigma_n^2 = 0.1^2$. The data batches contain 1, 5, 10, 25, 50 and 100 input trajectories, each having 25 data points. We repeat the experiments 50 times and report the average error.

The left plot in Figure 4 illustrates the cumulative discrepancy in the state distributions over time, and the right plot shows the error between the true and estimated drift/diffusion. Unsurprisingly, the discrepancy in both plots decrease when more training data is used. We also observe that the performance gain is insignificant after 50 input trajectories.

3.3. Ice core data

As another showcase of our model, we consider the NGRIP ice core dataset [22], which contains records of isotopic oxygen δ^{18}O concetrations within clacial ice cores. The record is used to explore the climatic changes that date back to last glacial period. During that period, the North Atlantic region underwent abrupt climate changes known as Dansgaard-Oeschger (DO) events. The events are characterized by a sudden increase in the temperature followed by a gradual cooling phase. Following [23], we consider $N = 2000$ timepoints from the time span from 60000 years to 20000 years before present, where 16 DO events have been identified.

Figure 5(a) illustrates the highly variable data. We observe a repeating pattern of DO events: a sudden increase followed by a slower settlement phase. The panel 5(b) indicates estimated drift that pushes the oxygen down until state -41 and up with small states, matching the data. Interestingly, diffusion at 5(c) is highly peaked between -42 and -43, which has accurately identified the regime of DO events. The model has learned to explain the DO events with high diffusion.

3.4. Human motion dataset

We finally demonstrate our approach on human motion capture data. Our goal is twofold: to estimate a single drift function that captures the dynamics of the walking sequences of several people, and to explain the discrepancies among sequences via diffusion. The input trajectories are the same as in [24]: four walking sequences, each from a different person. We also follow the preprocessing method described in [24], which results in a simplified skeleton that consists of 50-dimensional pose configurations. All records are mean centered and downsampled by a factor of two.

Inference is performed in three dimensional space where the input sequences are projected using PCA. We place the inducing points on a $5 \times 5 \times 5$ grid and set the length-scale of both drift and diffusion process to 0.5. $3D$ data set, inferred drift fit and the density of the sample paths are visualized in Figure 6. We can conclude that our model is capable of inferring drift and diffusion functions that match arbitrary data.

4. DISCUSSION

We propose an approach for learning non-parametric drift and diffusion functions of stochastic differential equation (SDE) systems such that the resulting simulated state distributions match data. Our approach can learn arbitrary dynamics due to the flexible inducing Gaussian process formulation. We propose a stochastic estimate of the simulated state distributions and an efficient system of computing their gradients. Our approach does not place any restrictions on the sparsity or denseness of the observations data. We leave learning of time-varying drifts and diffusions as interesting future work.

Acknowledgements. The data used in this project was obtained from `mocap.cs.cmu.edu`. The database was created with funding from NSF EIA-0196217. This work has been supported by the Academy of Finland Center of Excellence in Systems Immunology and Physiology, the Academy of Finland grants no. 260403, 299915, 275537, 311584.

5. REFERENCES

[1] R. Friedrich, J. Peinkeb, M. Sahimic, and R. Tabar, "Approaching complexity by stochastic methods: From biological systems to turbulence," *Phys. reports*, vol. 506, pp. 87–162, 2011.

[2] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*, Springer, 6th edition, 2014.

[3] A. Ruttor, P. Batz, and M. Opper, "Approximate Gaussian process inference for the drift function in stochastic differential equations," in *Advances in Neural Information Processing Systems*, 2013, pp. 2040–2048.

[4] C. García, A. Otero, P. Felix, J. Presedo, and D. Marquez, "Nonparametric estimation of stochastic differential equations with sparse Gaussian processes," *Physical Review E*, vol. 96, no. 2, pp. 022104, 2017.

[5] C. Archambeau, D. Cornford, M. Opper, and J. Shawe-Taylor, "Gaussian process approximations of stochastic differential equations," in *Gaussian Processes in Practice*, 2007.

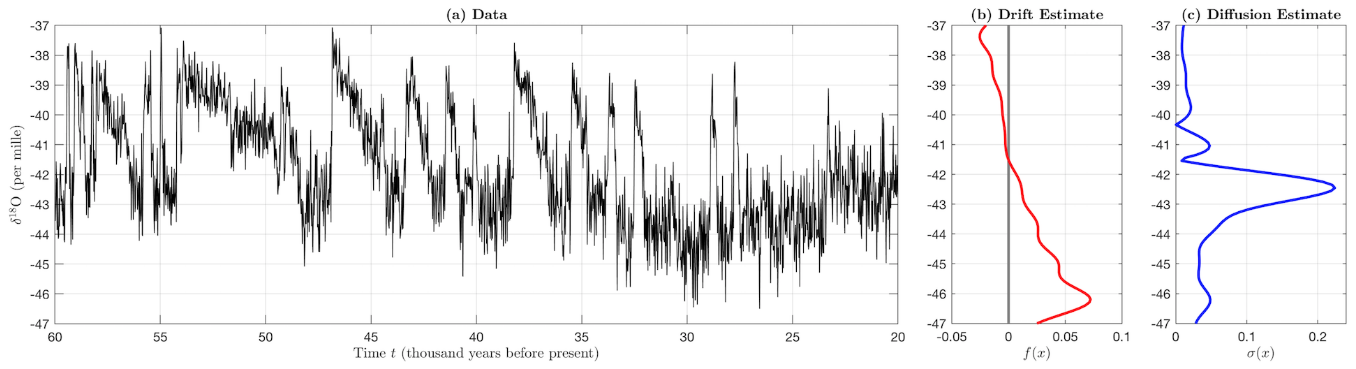

Fig. 5. **(a)** The ice core data, **(b)** estimated drift, **(c)** the highly state-dependent diffusion.

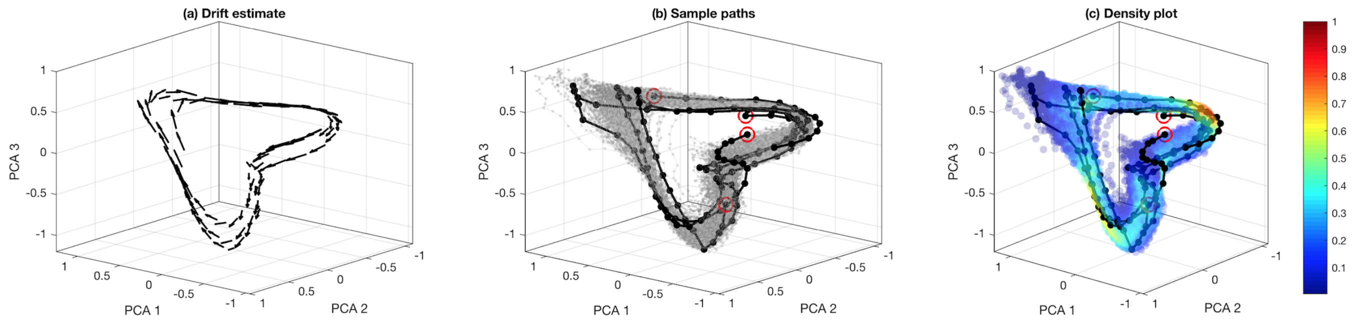

Fig. 6. **(a)** Shared drift estimate learned from walking data of 4 subjects, **(b)** estimated sample paths, **(c)** density plots of the sample paths. The 4 observed trajectories are shown as black lines in **(b-c)**, with red circles denoting the initial state.

[6] M. Vrettas, M. Opper, and D. Cornford, "Variational mean-field algorithm for efficient inference in large systems of stochastic differential equations," *Physical Review E*, vol. 91, pp. 012148, 2015.

[7] P. Batz, A. Ruttor, and M. Opper, "Approximate bayes learning of stochastic differential equations," *arXiv:1702.05390*, 2017.

[8] C.E. Rasmussen and K.I. Williams, *Gaussian processes for machine learning*, MIT Press, 2006.

[9] J. M. Varah, "A spline least squares method for numerical parameter estimation in differential equations," *SIAM J sci Stat Comput*, vol. 3, pp. 28–46, 1982.

[10] S. Ellner, Y. Seifu, and R. Smith, "Fitting population dynamic models to time-series data by gradient matching," *Ecology*, vol. 83, pp. 2256–2270, 2002.

[11] J. Ramsay, G. Hooker, D. Campbell, and J. Cao, "Parameter estimation for differential equations: a generalized smoothing approach," *J R Stat Soc B*, vol. 69, pp. 741–796, 2007.

[12] M. Heinonen, C. Yildiz, H. Mannerström, J. Intosalmi, and H. Lähdesmäki, "Learning unknown ODE models with Gaussian processes," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[13] M. Alvarez, L. Rosasco, and N. Lawrence, "Kernels for vector-valued functions: A review," *Foundations and Trends in Machine Learning*, 2012.

[14] N. Wahlström, M. Kok, and T. Schön, "Modeling magnetic fields using Gaussian processes," *IEEE ICASSP*, 2013.

[15] I. Macedo and R. Castro, "Learning divergence-free and curl-free vector fields with matrix-valued kernels," *Instituto Nacional de Matematica Pura e Aplicada*, 2008.

[16] C. Micchelli and M. Pontil, "On learning vector-valued functions," *Neural computation*, 2005.

[17] J. Quiñonero-Candela and C.E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[18] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*, Cambridge University Press, 2004.

[19] J. Hensman, N. Fusi, and N. Lawrence, "Gaussian processes for big data," in *Uncertainty in Artificial Intelligence*. AUAI Press, 2013, pp. 282–290.

[20] P. Kokotovic and J. Heller, "Direct and adjoint sensitivity equations for parameter optimization," *IEEE Trans. on Automatic Control*, vol. 12, pp. 609–610, 1967.

[21] F. Fröhlich, B. Kaltenbacher, F. Theis, and J. Hasenauer, "Scalable parameter estimation for genome-scale biochemical reaction networks," *PLOS Comp Biol*, vol. 13, pp. 1–18, 2017.

[22] K. Andersen, N. Azuma, J-M. Barnola, M. Bigler, P. Biscaye, N. Caillon, J. Chappellaz, H. Clausen, et al., "High-resolution record of northern hemisphere climate extending into the last interglacial period," *Nature*, vol. 431, pp. 147, 2004.

[23] F. Kwasniok, "Analysis and modelling of glacial climate transitions using simple dynamical systems," *Phil Trans R Soc A*, vol. 371, pp. 20110472, 2013.

[24] J. Wang, D. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Trans. on pattern analysis and machine intelligence*, vol. 30, pp. 283–298, 2008.