ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Scalable Variational Inference for Stochastic Differential Equations

Master Thesis

Ruifeng Xu

September 1, 2017

Supervisor: Prof. Dr. Joachim M. Buhmann

Advisors: Stefan Bauer & Nico S. Gorbach

Department of Computer Science, ETH Zürich

Abstract

State and parameter estimation in dynamical systems based on sparse, discrete observations is a topical yet challenging problem. Traditional methods suffer from extremely high computational costs due to the need to carry out explicit numerical integration after parameter adaptation. In contrast, the recently proposed gradient matching with Gaussian processes model is a promising tool. It is a grid-free inference technique that also eliminates the dependency on numerical integration. However, due to the intractability of the posterior, approximate inference techniques must be used. Sampling-based solutions fall short in this case since most real-world dynamical systems are high-dimensional. On the other hand, variational approaches have shown their potential both in terms of prediction accuracy and runtime performance, even in situations when the system is only partially observed. Extending the state-of-art variational gradient matching framework, this work further improves the flexibility of the inference algorithm by relaxing the structural assumption on the dynamical systems. Support for positivity constraints on the states and parameters that are common in many biochemical and physical applications is also introduced. Finally, a highly efficient parallel solution is devised to address problems involving stochastic differential equations.

Contents

iv

Chapter 1

Introduction

Modelling complicated natural phenomena using mathematical abstractions is a common practice in modern scientific research and real-world applications. A successful model of a phenomenon helps us interpret its key features while omits extraneous details (Babtie et al., 2014). For various dynamic processes underlying a broad range of fields including chemistry, physics, biology, economics, meteorology, etc., one convenient and useful abstraction is to describe them as a set of *ordinary differential equations (ODEs)* or *stochastic (ordinary) differential equations (SDEs)* (Ellner and Guckenheimer, 2011; Gardiner, 2009), which are also called *deterministic dynamical systems* and *random dynamical systems* respectively.

Using the term *dynamical system* to refer to the above two systems in general, this work addresses the topical yet challenging problem of statistical inference of the states and parameters of dynamical systems given noisy, sparse or even incomplete states observation. By extending the state-of-art approaches that combine the techniques of Gaussian process regression, gradient matching, and variational inference, the aim is to build an inference pipeline that operates efficiently, predicts accurately and scales to large systems.

1.1 Motivation

To understand the importance of the problem, it is possible to find numerous alternative models that describe the observations unless there is enough domain expertise to designate a specific model (Babtie et al., 2014). In order to select a good candidate, each of the alternatives must be trained using the observations, and then the results will be evaluated according to certain criteria. Even if the general underlying process is known, the parameters that control the dynamics of the system still need to be inferred from the observations. It is therefore critical for the data fitting process to be accu-

rate and robust. Meanwhile, the procedure should also be performant so that complex models can be trained within reasonable time constraints. For more concrete illustration of modelling using dynamical systems, consider the following two examples.

Deterministic dynamical systems One essential task in system biology is to compare and select an appropriate model to characterize a biochemical system, e.g. protein signalling transduction pathway (Vyshemirsky and Girolami, 2007). Typically, the structure of the system can be viewed as a network of biochemical reactions, which can be formally described by a group of nonlinear ODEs; the transitions among the network components, e.g. protein spieces, are determined by a set of kinetic parameters (Macdonald and Husmeier, 2015). During experiments, only concentrations on the species over a time span are observed. Therefore, inference of the true parameter values or even the *a posteriori* distributions for different models given the experimental data is of crucial importance for model selection purposes.

Random dynamical systems In reality, it is often that the mathematical model is not able to capture all the characteristics of the phenomenon. If we allow some randomness inside a deterministic dynamical system, we would get a random dynamical system (Øksendal, 2013). An example where random dynamical systems have a long history of application is weather forecasting, where the continuous evolution of the atmosphere is described by discretized quantities like pressure, temperature, wind speed, etc., measured at fixed intervals (Archambeau et al., 2007). Therefore, it is reasonable to model the rest of the unknown dynamics as the noise process within the SDEs. For such systems, useful prediction depends not only on the realistic modelling of the atmospheric environment, but also on the precise estimation of the initial conditions (Kalnay, 2003), due to the high sensitivity of future states on the initial conditions that is also known as the famous *butterfly effect* (Lorenz, 2000).

To summarize, the relative importance of state versus parameter estimation varies, depending on the specific application. State estimation is more important for short-term predictions such as weather forecasting, while parameter estimation is more important for long-term targets such as climate pattern modelling (Vrettas et al., 2015).

This work places more emphasis on the inference of system parameters since they shed light on the internal mechanisms of a dynamical system. Nevertheless, the states are also simultaneously estimated as a consequence of the design of the algorithm, which will be shown in later chapters.

1.2 Challenges

Although solving statistical inference problems involving ODEs and SDEs is useful in practice, but from a technical point of view, such problems involve many technical challenges as below.

First, since closed-form solutions do not exist for most ODEs and not at all for SDEs, conventional methods based on explicit numerical integrations are computationally expensive, which renders them impractical even for small-sized applications.

Second, the likelihood surfaces in the parameter space are likely multimodal due to nonlinearity within the dynamical systems (Calderhead et al., 2009), and may exhibit many local maxima, which makes the parameter searching challenging. From a Bayesian perspective, the intractability of the marginalization term is conventionally approximated using *Markov chain Monte Carlo (MCMC)* sampling schemes, which are in general very flexible but come at the cost of high computational intensity and onerous convergence analysis. Hence, the applicability of sampling-based approaches is largely subject to the dimensionality of the system under inference (Vrettas et al., 2015).

Last, in most of the realistic scenarios, only corrupted, and sometimes even only sparse and partial observations are available. Devising inference algorithms that are robust in such situations is itself a difficult topic across all machine learning paradigms.

1.3 Contributions

The contributions of this work are as follows. The Laplace mean-field approximation is proposed to relax the structural assumption on the dynamical systems. Through reparameterization on the optimization objectives, positivity constraints on the states and parameters are supported, which are essential for many real-world application. By utilizing the Doss-Sussmann/ Imkeller-Schmalfuss correspondence, the Laplace mean-field approximation is further extended to devise a distributed inference method to infer states and parameters of the SDEs. A brand new Python based solution is also implemented.

1.4 Organization

This thesis is organized as follows. Chapter 2 gives a general introduction to deterministic and random dynamical systems and provides a brief review of other related work on inference in dynamical systems. Chapter 3 introduces in detail the gradient matching with Gaussian process framework

(Calderhead et al., 2009; Dondelinger et al., 2013) and its recent improvement (Gorbach et al., 2016, 2017) based on variational inference, which are the foundation of this work. In Chapter 4, the Laplace approximation technique is applied to the variational gradient matching model to derive a new solution that relaxes the structural assumption about the ODEs, and further introduces positivity constraints on the states and parameters through reparameterization. In Chapter 5, an ensemble-like inference solution for SDEs is derived by using the Doss-Sussmann/Imkeller-Schmalfuss correspondence. The examination and comparison of the accuracy, performance and scalability of the solutions proposed in this work are conducted in Chapter 6. Lastly, Chapter 7 draws the conclusion.

Chapter 2

Inference in Dynamical Systems

This chapter reviews the necessary basics about ODEs (Section 2.1) and SDEs (Section 2.2) with an emphasis on the task of inferring states and parameters of dynamical systems given noisy, sparse or even incomplete state observations. Comprehensive discussions about ODEs and SDEs together with their numerical solutions can be found in textbooks such as Butcher (2016) and Øksendal (2013). Furthermore, the notations used to describe the dynamical systems, and the model under noisy observations are also introduced, which will be used throughout this work. In Section 2.3, a survey of related work is provided. The details about the inference technique using variational gradient matching with Gaussian processes, which are the core foundations of this work, are examined in Chapter 3.

2.1 Deterministic dynamical systems

A K-dimensional deterministic dynamical system, i.e. a system with K states, can be described by a set of ODEs as follows:

$$\dot{\boldsymbol{x}}(t) = \frac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{\theta}) \tag{2.1}$$

where $\boldsymbol{x}(t) = [x_1(t), \ldots, x_K(t)]^\top \in \mathbb{R}^K$ is a vector containing the K states of the system at time point t with their time derivatives collectively denoted by $\dot{\boldsymbol{x}}(t) = \frac{d\boldsymbol{x}(t)}{dt} = [f_1(\boldsymbol{x}(t), \boldsymbol{\theta}), \ldots, f_K(\boldsymbol{x}(t), \boldsymbol{\theta})]^\top \in \mathbb{R}^K$, and $\boldsymbol{f} : \mathbb{R}^K \mapsto \mathbb{R}^K$ encodes the functional relationship between the states and their derivatives over time that is in turn governed by a vector of parameters $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_M]^\top \in \mathbb{R}^M$. Note that in general, $\boldsymbol{f}$ may have direct dependency on time, which is suppressed here to simplify the notations.

From Butcher (2016), the specification of the ODEs alone is not interesting since it generally does not guarantee a unique solution. But if the *initial condition* $\boldsymbol{x}(t_0) = [x_1(t_0), \ldots, x_K(t_0)]^\top$ is given, then together with (Eq. 2.1), they

define a problem known as the *initial value problem*, where the goal is to solve the differential equations to approximate the future states of the dynamical system. Three important aspects about the initial value problem are the existence of a solution, the uniqueness of the solution, and the sensitivity of the solution due to small perturbations to the initial condition.

Within the context of this work, the initial problem can be therefore specified as the estimation of the states and the parameters of the ODEs based on state observations that are usually contaminated by noise. In light of this, the following paragraphs introduce the notations and the probabilistic model of a dynamical system under noisy observations, which will be used later in this work.

Noisy observation model

Suppose for the K-dimensional deterministic dynamical system given by (Eq. 2.1), we have a sequence of noisy observations $Y = [y(t_1), \ldots, y(t_N)] \in \mathbb{R}^{K \times N}$ over N time points, whose the corresponding true states values are $X = [x(t_1), \ldots, x(t_N)] \in \mathbb{R}^{K \times N}$ such that $y(t_n) = x(t_n) + \varepsilon(t_n)$ for $n = 1, \ldots, N$, where $\varepsilon(t_n) \in \mathbb{R}^K$ denotes the observation noises for the K states $x(t_n)$ at time point t_n. The above description can be succinctly written in matrix notation as

$$Y = X + E$$

where Y and X are defined before, and $E = [\varepsilon(t_1), \ldots, \varepsilon(t_N)] \in \mathbb{R}^{K \times N}$.

For simplicity, we assume that the observation noises $\varepsilon(t_n)$ at each time point are additive and state specific. Moreover, they follow an *independent and identically distributed (i.i.d.)* multivariate Gaussian distribution with zero mean and a diagonal covariance matrix across all time points, i.e. $\varepsilon_{(.)} \sim \mathcal{N}(\mathbf{0}, \mathbf{D})$, where $D_{ik} = \delta(i, k)\sigma_k^2$ for $i, k = 1, \ldots, K$, δ is the *Kronecker delta* function, and σ_k^2 is the variance of the observation noise for state k.

Let $y_k(t_n) \in \mathbb{R}$ and $x_k(t_n) \in \mathbb{R}$ be the observation and true value for the k-th state at time point t_n respectively. We can then collectively use $y_k = [y_k(t_1), \ldots, y_k(t_N)]^\top \in \mathbb{R}^N$ and $x_k = [x_k(t_1), \ldots, x_k(t_N)]^\top \in \mathbb{R}^N$ to denote the sequence of observations and the true values for the k-th state over the N time points. With the above noise assumption and notations, we have

$$p(Y|X, \sigma) = \prod_k \mathcal{N}(y_k | x_k, \sigma_k^2 I) \tag{2.2}$$

Note that the discussion within this work can be generalized to cases where only combinations of states are observed such that $Y = HX + E$, where H describes the relationship between the states and the observations. In order

to simplify the notation, we assume that H is equal to the identity matrix I. Furthermore, handling of the cases where some states are not observed is also possible and will be presented in the relevant sections later.

2.2 Random dynamical systems

As another important family of dynamical systems, random dynamical systems, also referred to as *diffusion processes*, have been widely applied to various domains because of their capability to incorporate unknown processes as internal noise processes (Vrettas et al., 2011). On a high level, a random dynamical system is a continuous time *Markov process* consisting of a deterministic part and a stochastic noise driven component (Riesinger et al., 2016). Such system is described by a set of SDEs and requires the special *stochastic calculus*.

Based on Øksendal (2013) and Vrettas et al. (2015), a *K*-dimensional SDE system defined on a probability space $(\mathbf{\Omega}, \mathcal{F}, \mathbb{P})$ is represented in the *Itô* form as

$$dx(t) = f(x(t), \theta)dt + g(x(t), \rho)dW_t \tag{2.3}$$

where $f : \mathbb{R}^K \mapsto \mathbb{R}^K$ is the deterministic *drift function* with *drift parameter* vector $\theta = [\theta_1, \ldots, \theta_M]^\top \in \mathbb{R}^M$, $g : \mathbb{R}^k \mapsto \mathbb{R}^{K \times W}$ is the coefficient function for the noise process with *diffusion parameter* vector $\rho = [\rho_1, \ldots, \rho_V]^\top \in \mathbb{R}^V$, and $dW_t = [dW_t^1, \ldots, dW_t^W]^\top \in \mathbb{R}^W$ is the differential of a standard *Wiener process* W_t of dimension W, i.e. $dW_t \sim \mathcal{N}(\mathbf{0}, dt\mathbf{I})$. In its integral form, the above equation is equivalent to

$$x(t_T) = x(t_0) + \int_0^T f(x(t_s), \theta)ds + \sum_i^W \int_0^T g_i(x(t_s), \rho)dW_s^i \tag{2.4}$$

where $x(t_0)$ denotes the initial condition and $g_i(x(t_s), \rho)$ is the *i*-th column of the corresponding noise coefficient matrix. As W_t is a stochastic process, each time we solve the above integration, we would obtain mostly likely a different *sample path*. Note that similar to the ODEs in (Eq. 2.1), the dependency of f and g on time t is suppressed here to unclutter the notations.

The above equations define a linear diffusion process with multiplicative noises. For simplicity, we consider only stochastic systems with state-specific, additive white noise in this work. A class of multiplicative noise models can be mapped to this model through reparametrization (Vrettas et al., 2011). This assumption means that we can simplify the SDEs to

$$dx(t) = f(x(t), \theta)dt + \Sigma^{\frac{1}{2}}dW_t \tag{2.5}$$

where Σ is the diagonal noise covariance matrix, i.e. $\Sigma_{ik} = \delta(i,k)\rho_k^2$ for $i, k = 1, \ldots, K$, and W_t becomes a standard K-dimensional Wiener process.

Skipping over the details of stochastic calculus, here we give an intuitive understanding of the evolution of the system by using the *Euler-Maruyama* (Higham, 2001) representation as follows:

$$x(t_{n+1}) - x(t_n) = f(x(t_n), \theta)\Delta t + \sqrt{\Delta t \Sigma} \varepsilon_{t_n} \tag{2.6}$$

where $\Delta t = t_{n+1} - t_n$ denotes the time increment and ε_{t_n} is a standard multivariate Gaussian random vectors, i.e. $\varepsilon_{t_n} \sim \mathcal{N}(0, I)$. The SDEs in (Eq. 2.5) can be considered as the limit of the process described by (Eq. 2.6) (Archambeau et al., 2007).

Without redundancy, the noisy observation model and the notations introduced previously in Section 2.1 can be directly applied to the SDE models in the rest of this text.

2.3 Related work

For state and parameter inference in ODEs, *gradient matching* has established itself in recent years as a promising tool. The main idea behind gradient matching is straightforward. It first interpolates the states X from the observations Y using a smoothing technique, and then minimizes the discrepancy between slopes of the interpolants and the derivatives obtained from the ODEs $f(x(t), \theta)$ (Macdonald and Husmeier, 2015). This dramatically improves the inference efficiency since the ODEs never need to be solved explicitly in contrast to traditional techniques.

Early applications of gradient matching trace back to the spline based method proposed by Ramsay et al. (2007). Unfortunately, such a solution requires full observability of the system and is difficult to extend. More recently, gradient matching with Gaussian processes (GMGP) (Calderhead et al., 2009; Dondelinger et al., 2013) model has been proposed. The introduction of Gaussian processes makes the inference technique grid free and is even capable of handling partially observed systems. In the original GMGP model, sampling is used to estimate the intractable posterior. A more performant and scalable solution called variational GMGP (VGMGP) is proposed by Gorbach et al. (2017), which uses variational technique. As the foundation of this work, details of the original GMGP method and the VGMGP method are discussed in Chapter 3.

For inference problems involving random dynamical systems, classical approaches typically resort to Kalman filters and MCMC. The MCMC based solutions scale poorly in practice. As another approximate inference technique, variational methods have gained popularity in recent years. For example, the variational Gaussian process approximation (VGPA) proposed by

Archambeau et al. (2007) uses a linear approximation strategy to infer the states and the parameters of diffusions processes. It consists of two steps. In the forward step, the mean and covariance of the approximate process are estimated. Then in the backward step, the time evolution of the Lagrange multipliers are calculated. The Lagrange multipliers ensure consistency for the mean and covariance. These two steps are then iteratively executed to improved the result. Further improvements based on this method can be found in Archambeau et al. (2008) and Vrettas et al. (2011, 2015).

Chapter 3

Gradient Matching with Gaussian Processes

3.1 Preliminary

This section briefly introduces Gaussian process regression (Section 3.1.1), Kullback-Leibler (KL) divergence (Section 3.1.2), and mean-field variational inference (Section 3.1.3) as the essential machine learning topics for the rest of the work. Formal treatment can be found by following the references listed for each topic.

3.1.1 Gaussian process regression

As a nonparametric, kernel-based Bayesian inference technique, Gaussian process regression is a powerful tool for many non-linear regression tasks. In terms of application, a Gaussian process prior needs first to be specified on the regression function $f(x)\colon \mathbb{R}^D \to \mathbb{R}$. After observing data, the prior can be turned into a posterior distribution to make predictions (Murphy, 2012). The definition of Gaussian process given by Rasmussen and Williams (2006) is listed as follow.

Definition 3.1 *A Gaussian process is a collection of random variables such that any finite subset of it forms a multivariate Gaussian distribution.*

The Gaussian process prior on f is denoted as

$$f(x) \sim \mathcal{GP}(m(x),\, k(x,\, x')) \tag{3.1}$$

where

$$m(x) = \mathbb{E}[f(x)] \tag{3.2}$$
$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))] \tag{3.3}$$

The above equations simply mean that for any finite collection of points $X = \{x_i \in \mathbb{R}^D | i = 1, \ldots, N\}$, we have $f|X \sim \mathcal{N}(m, K(X, X))$, where $f_i = f(x_i)$, $m_i = m(x_i)$, and $K_{ij} = k(x_i, x_j)$ for $i, j = 1, \ldots, N$. This shows that a Gaussian process is completely specified by its *mean function* (Eq. 3.2) and *covariance function* (Eq. 3.3). Furthermore, to form a multivariate Gaussian distribution, the covariance function must generate a covariance matrix that is positive definite for any set of points.

We often only have access to noisy observations denoted by $y = \{y_i | i = 1, \ldots, N\}$ such that $y_i = f(x_i) + \varepsilon_i$ for $i = 1, \ldots, N$. We assume i.i.d. additive noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. For any finite collection of test points $X^* = \{x_i^* \in \mathbb{R}^D | i = 1, \ldots, M\}$, we have

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m \\ m^* \end{bmatrix}, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix} \right)$$

The posterior on f^* can then be easily found in closed-form using the standard conditional Gaussian distribution formula to obtain

$$f^* | X, y, \sigma, X^* \sim \mathcal{N}(m^* + K(X^*,X)[K(X,X) + \sigma^2 I]^{-1}(y - m),$$
$$K(X^*,X^*) - K(X^*,X)[K(X,X) + \sigma^2 I]^{-1}K(X,X^*)) \quad (3.4)$$

Note that in practice, the mean function is usually assumed to be zero, i.e. $m(x) = 0$ (Rasmussen and Williams, 2006).

3.1.2 Kullback-Leibler divergence

Suppose there are two continuous[1] probability distributions p and q, and the task is to approximate the unknown distribution p using any distribution q. In information theory, a frequently used dissimilarity measure between p and q is the *forward KL divergence* or *relative entrophy* (Kullback and Leibler, 1951), which is defined as

$$KL(p\|q) = \int p(x) \ln \frac{p(x)}{q(x)} dx \quad (3.5)$$
$$= \int p(x) \ln p(x) dx - \int p(x) \ln q(x) dx$$
$$= -H(p) + H(p,q)$$

where $H(P)$ is the *entropy* of the distribution p, and $H(p,q)$ is the *cross entrophy* between p and q.

---

[1]Given the context of this work, the discussion here focuses more on continuous probability distributions. However similar reasoning and results can be applied to discrete probability distributions as well.

Similarly, the *reverse KL divergence* $KL(q\|p)$ can be defined as

$$
\begin{aligned}
KL(q\|p) &= \int q(\boldsymbol{x}) \ln \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})} d\boldsymbol{x} \\
&= \int q(\boldsymbol{x}) \ln q(\boldsymbol{x}) d\boldsymbol{x} - \int q(\boldsymbol{x}) \ln p(\boldsymbol{x}) d\boldsymbol{x} \\
&= -H(q) + H(q, p)
\end{aligned}
\tag{3.6}
$$

To see why the KL divergence can be used as a dissimilarity measure, one can use *Jensen's inequality* (Jensen, 1906) to prove the following stated as a theorem. The proof (Bishop, 2006, Section 1.6) is omitted here. However, it is not a distance measure since $KL(p\|q) \neq KL(q\|p)$ in general, i.e. it is not symmetric (Goodfellow et al., 2016).

Theorem 3.2 $KL(p\|q) \geqslant 0$ *with* $KL(p\|q) = 0$ *if and only if* $p(\boldsymbol{x}) = q(\boldsymbol{x})$ *almost everywhere. Similar argument also applies to* $KL(q\|p)$.

From the above, the proxy distribution q can therefore be found by minimizing either $KL(p\|q)$ (Eq. 3.5) or $KL(q\|p)$ (Eq. 3.6) . However, the effects of the two minimization strategies are different. The intuition behind is summarized here from the discussions in Bishop (2006) and Murphy (2012).

- The minimization of $KL(p\|q)$ can be characterized as *zero avoiding* for q because there is a large contribution to the integral at places where $p(\boldsymbol{x}) > 0$ and $q(\boldsymbol{x}) = 0$. As a result, we need to ensure that $q(\boldsymbol{x}) > 0$ whenever $p(\boldsymbol{x}) > 0$, which generally leads to an overestimation of the support for p.

- Conversely, the minimization of $KL(q\|p)$ can be viewed as *zero forcing* for q due to the large contribution to the integral at place where $p(\boldsymbol{x}) = 0$ and $q(\boldsymbol{x}) > 0$. Consequently, we need to force $q(\boldsymbol{x}) = 0$ whenever $p(\boldsymbol{x}) = 0$, which tends to underestimate the support for p.

Visual illustrations of approximating both a single 2-D Gaussian distribution and mixture of two 2-D Gaussian distributions using another 2-D Gaussian distribution can be found in Section 10.1 of Bishop (2006) and Section 21.2 of Murphy (2012).

3.1.3 Mean-field variational inference

The evaluation of the posterior distribution given the observations plays a central role in Bayesian statistics since it is essential to make point estimates, form predictive distributions, etc. (Bishop, 2006). Unfortunately, solutions to many of such problems are not analytically tractable, e.g. the Bayesian mixture of Gaussians example in Section 2.1 of Blei et al. (2017), and hence, we have to resort to approximation algorithms. Here, we review *variational*

inference (Jordan et al., 1999; Wainwright et al., 2008), which is a deterministic approximation scheme that approximates probability densities through optimization. Compared to the stochastic MCMC methods, variational inference tends to be faster and has been shown empirically to be capable of scaling to large datasets, for example topic modeling using 1.8 million New York Times articles (Hoffman et al., 2013), traffic pattern analysis using 1.7 million taxi rides (Kucukelbir et al., 2017). The core concepts behind variational inference are summarized below based on Bishop (2006) and Blei et al. (2017).

Evidence lower bound

Suppose given the model specification $p(x, z)$, we want to approximate the unknown posterior $p(z|x)$, where z denotes the *latent (hidden) variables*[2], and x denotes the observed variables. In variational inference, we first posit a family of distributions $\mathcal{Q}$ over z parameterized by a set of free *variational parameters*[3]. The "closest" approximation $q(z)$ to $p(z|x)$ can then be found by minimizing the KL divergence $KL(q(z)\|p(z|x))$.

$$q^*(z) \in \underset{q(z) \in \mathcal{Q}}{\arg\min} \, KL(q(z)\|p(z|x)) \tag{3.7}$$

However, directly solving the above optimization problem is infeasible since expanding the objective reveals its dependency on the *evidence* $p(x)$, and we assume that $p(x) = \int p(x, z) dx$ is intractable.

$$
\begin{aligned}
KL(q(z)\|p(z|x)) &= \int q(z) \ln q(z) dz - \int q(z) \ln p(z|x) dz \\
&= \int q(z) \ln q(z) dz - \int q(z) \ln p(x, z) dz + \ln p(x)
\end{aligned}
\tag{3.8}
$$

Instead, we can rearrange the terms in (Eq. 3.8) to obtain

$$\ln p(x) = KL(q(z)\|p(z|x)) + \mathcal{L}(q(z)) \tag{3.9}$$

where

$$\mathcal{L}(q(z)) = -\int q(z) \ln q(z) dz + \int q(z) \ln p(x, z) dz \tag{3.10}$$

Because $KL(q(z)\|p(z|x)) \geqslant 0$ (Theorem 3.2), $\mathcal{L}(q(z))$ can be treated as a lower bound on $\ln p(x)$, i.e. $\ln p(x) \geqslant \mathcal{L}(q(z))$, and hence, it is also called

---

[2]In full-Bayesian treatment, other model parameters, when they exist, are absorbed into z to simplify the notation.

[3]The dependency on the variational parameters is left implicit in the notations used in the following text.

the *evidence lower bound*. Furthermore, since $\ln p(x)$ is a constant, we can state that *maximizing $\mathcal{L}(q(z))$ is equivalent to minimizing $KL(q(z)\|p(z|x))$*.

Rewriting (Eq. 3.10) as (Eq. 3.11) below also provides us with some intuitions about the form of the optimal proxy distribution. The first term in (Eq. 3.11) encourages $q(z)$ to put weights on z that explains x through $p(x|z)$, while the second term encourages $q(z)$ to be close to the prior $p(z)$. This reflects the usual balance between likelihood and prior in Bayesian statistics.

$$\mathcal{L}(q(z)) = -\int q(z)\ln q(z)dz + \int q(z)\ln\left[p(x|z)p(z)\right]dz$$
$$= \mathbb{E}_q[\ln p(x|z)] - KL(q(z)\|p(z)) \tag{3.11}$$

Mean-field variational family

Having specified the optimization objective in terms of $\mathcal{L}(q(z))$, we now need to decide on the family of distributions $\mathcal{Q}$. Specifically, we focus on the *mean-field variational family*, where the latent variables and model parameters are factorized mutually independent groups, and each group is controlled by its own variational parameters, i.e. $q(z) = \prod_i q_i(z_i)$. Note that since the complexity of the optimization is determined by the reach of $\mathcal{Q}$, one should be aware of the trade-off between model flexibility and runtime efficiency when choosing the reach of $\mathcal{Q}$

By dissecting out the terms that are dependent on $q_i(z_i)$, and absorbing the terms that are independent of it into constant from $\mathcal{L}(q(z))$, the optimal solution $q_i^*(z_i)$ can be expressed as

$$q_i^*(z_i) \propto \exp\left\{\mathbb{E}_{q,j\neq i}[\ln p(x,z)]\right\} \tag{3.12}$$

where $\mathbb{E}_{q,j\neq i}[\ln p(x,z)]$ denotes the expectation of the log of the joint distribution $p(x,z)$ with respect to all the other factors q_j except q_i. The derivation (Bishop, 2006, Section 10.1) is omitted here.

The advantage of the factorization assumption is that we can fix other factors and optimize only one factor at a time in an iterative manner until convergence, or the preconfigured maximum number of iterations has been reached. This approach is called the *coordinate ascent mean-field variational inference*. Detailed pseudocode for the algorithm, handling for numerical stability, test for convergence, etc. are discussed in Blei et al. (2017).

As a consequence of the optimzation objective and the factorization assumption, the optimal solution tends to underestimate the variance of the posterior to produce a too compact distribution. Discussion about this issue can be found in Section 3.1.2 of this work and Section 10.1 of Bishop (2006). Lastly, in cases where the objective is non-convex, convergence is only guaranteed to a local optimum subject to the initialization (Blei et al., 2017).

15

3.2 Sampling-based gradient matching with Gaussian processes

This section reviews a recently proposed framework to infer parameters and states of ODEs using gradient matching with Gaussian processes (GMGP), which originally appeared in Calderhead et al. (2009) and was extended by Dondelinger et al. (2013).

Adopting the notations from Section 2.1, for a *K*-dimensional dynamical system described by (Eq. 2.1) as:

$$\dot{x}(t) = \frac{dx(t)}{dt} = f(x(t), \theta) \tag{3.13}$$

Calderhead et al. (2009) introduce independent Gaussian process priors on each state *k* for $k = 1, \ldots, K$ such that

$$p(X|\varphi) = \prod_k \mathcal{N}(x_k|\mathbf{0}, C_{\varphi_k}) \tag{3.14}$$

where C_{φ_k} is the covariance matrix induced by the corresponding kernel function $\mathcal{K}_{\varphi_k}$ with the hyperparameter vector φ_k over the *N* time points.

States interpolation

If we assume the observation on the states is corrupted by noise described by (Eq. 2.2) as:

$$p(Y|X, \sigma) = \prod_k \mathcal{N}(y_k|x_k, \sigma_k^2 I) \tag{3.15}$$

the posterior distribution on the states X is obtained as

$$p(X|Y, \varphi, \sigma) = \frac{p(X|\varphi)p(Y|X, \sigma)}{\int p(X|\varphi)p(Y|X, \sigma)dX}$$
$$= \prod_k \mathcal{N}(x_k|\mu_k(y_k), \Sigma_k) \tag{3.16}$$

where[4] $\mu_k(y_k) = C_{\varphi_k}(C_{\varphi_k} + \sigma_k^2 I)^{-1}y_k$ and $\Sigma_k = \sigma_k^2 C_{\varphi_k}(C_{\varphi_k} + \sigma_k^2 I)^{-1}$.

Gaussian process response model

Because differentiation is a linear operation, the derivative of a Gaussian process is also a Gaussian process (Rasmussen and Williams, 2006, Section 9.4). Hence, the joint distribution of the states and their derivatives within

---

[4]Another equivalent derivation, as stated in Gorbach et al. (2017), is to define $\mu_k(y_k) = \sigma_k^{-2}\Sigma_k^{-1}y_k$, and $\Sigma_k = (\sigma_k^{-2}I + C_{\varphi_k}^{-1})^{-1}$.

a finite amount of time points is Gaussian. To illustrate, we write the joint distribution of x_k and $\dot{x}_k$ as follows:

$$\begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} C_{\varphi_k} & C'_{\varphi_k} \\ {}'C_{\varphi_k} & C''_{\varphi_k} \end{bmatrix})$$

where

$$C_{\varphi_k i,j} = \mathcal{K}_{\varphi_k}(t_i, t_j)$$
$$C'_{\varphi_k i,j} = \frac{\partial \mathcal{K}_{\varphi_k}(t_i, t_j)}{\partial t_j}$$
$${}'C_{\varphi_k i,j} = \frac{\partial \mathcal{K}_{\varphi_k}(t_i, t_j)}{\partial t_i}$$
$$C''_{\varphi_k i,j} = \frac{\partial^2 \mathcal{K}_{\varphi_k}(t_i, t_j)}{\partial t_i \partial t_j}$$

In other words, C_{φ_k} and C''_{φ_k} are the *auto-covariance* matrices for x_k and $\dot{x}_k$ respectively, and C'_{φ_k} and $'C_{\varphi_k}$ are the *cross-covariance* matrices between x_k and $\dot{x}_k$ and between $\dot{x}_k$ and x_k respectively. Therefore, the conditional distribution over the state derivatives $\dot{X}$ is given by

$$p(\dot{X}|X, \boldsymbol{\varphi}) = \prod_k \mathcal{N}(\dot{x}_k|\boldsymbol{m}_k, A_k) \tag{3.17}$$

where[5] $\boldsymbol{m}_k = {}'C_{\varphi_k} C_{\varphi_k}^{-1} x_k$ and $A_k = C''_{\varphi_k} - {}'C_{\varphi_k} C_{\varphi_k}^{-1} C'_{\varphi_k}$.

ODE response model

Since given certain states and parameters, the ODEs (Eq. 3.13) are the natural information source of the state derivates, if we assume state specific, additive and normally distributed errors between the true state derivatives and response from the ODEs using the states and parameters estimation results as input, we have

$$p(\dot{X}|X, \boldsymbol{\theta}, \boldsymbol{\gamma}) = \prod_k \mathcal{N}(\dot{x}_k|\boldsymbol{f}_k(X, \boldsymbol{\theta}), \gamma_k I) \tag{3.18}$$

where the vector γ contains the error variances for each state, i.e. $\gamma = [\gamma_1, \ldots, \gamma_K]^T \in \mathbb{R}^K$.

---

[5]The derivation from Dondelinger et al. (2013) is used here since there is a minor inconsistency in the original proposal from Calderhead et al. (2009).
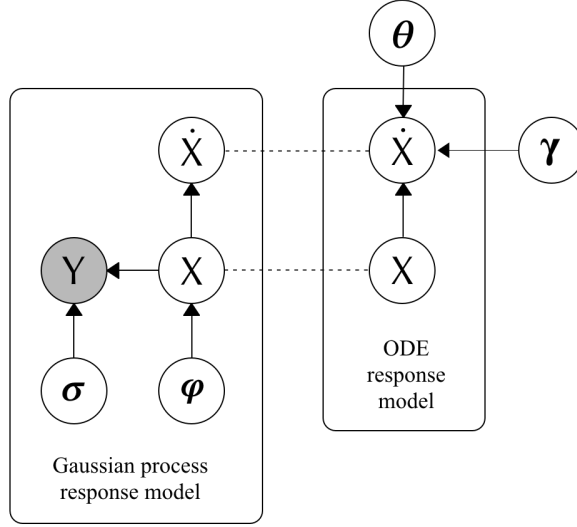
Figure 3.1: Graphical representation of the gradient matching with Gaussian processes model proposed by Calderhead et al. (2009). The dashed lines indicate information association from two response models using the product of experts (Eq. 3.19) . Details of the model are in Section 3.2.

Product of experts

Next, the *product of experts* (Hinton, 1999) approach is employed to combine the Gaussian process response model (Eq. 3.17) and the ODE response model (Eq. 3.18) to obtain

$$p(\dot{X}|X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma}) \propto p(\dot{X}|X, \boldsymbol{\varphi})p(\dot{X}|X, \boldsymbol{\theta}, \boldsymbol{\gamma}) \tag{3.19}$$

This product form implies that $p(\dot{X}|X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma})$ attains high densities at locations where both $p(\dot{X}|X, \boldsymbol{\varphi})$ and $p(\dot{X}|X, \boldsymbol{\theta}, \boldsymbol{\gamma})$ have strong support.

Full model distribution

To summarize, the full joint distribution of the GMGP model is graphically represented in Figure 3.1, which corresponds to the following:

$$p(Y, X, \dot{X}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \sigma, \boldsymbol{\gamma}) = p(Y|X, \sigma)p(X|\boldsymbol{\varphi})p(\dot{X}|X, \boldsymbol{\theta}, \boldsymbol{\varphi}, \boldsymbol{\gamma})p(\boldsymbol{\varphi})p(\boldsymbol{\theta})p(\sigma)p(\boldsymbol{\gamma})$$
$$\propto p(Y|X, \sigma)p(X|\boldsymbol{\varphi})p(\dot{X}|X, \boldsymbol{\varphi})p(\dot{X}|X, \boldsymbol{\theta}, \boldsymbol{\gamma})p(\boldsymbol{\varphi})p(\boldsymbol{\theta})p(\sigma)p(\boldsymbol{\gamma})$$

where $p(\boldsymbol{\varphi})$, $p(\boldsymbol{\theta})$, $p(\sigma)$, and $p(\boldsymbol{\gamma})$ denote the corresponding priors.

Original sampling scheme

Calderhead et al. (2009) propose to integrate over the latent state derivatives $\dot{X}$ to obtain

$$p(\boldsymbol{\theta}, \boldsymbol{\gamma}|X, \boldsymbol{\varphi}) = \int p(\boldsymbol{\theta})p(\boldsymbol{\gamma})p(\dot{X}|X, \boldsymbol{\theta}, \boldsymbol{\varphi}, \boldsymbol{\gamma})d\dot{X}$$

$$\propto \frac{p(\boldsymbol{\theta})p(\boldsymbol{\gamma})}{\mathcal{Z}(\boldsymbol{\gamma})} \exp\left[-\frac{1}{2}\sum_k (\boldsymbol{f}_k - \boldsymbol{m}_k)^T \boldsymbol{\Lambda}_k (\boldsymbol{f}_k - \boldsymbol{m}_k)\right] \qquad (3.20)$$

where $\boldsymbol{\Lambda}_k^{-1} = A_k + \gamma_k I$, $\mathcal{Z}(\boldsymbol{\gamma}) = \prod_k |2\pi(A_k + \gamma_k I)|^{\frac{1}{2}}$ and $\boldsymbol{f}_k = \boldsymbol{f}_k(X, \boldsymbol{\theta})$.

Then the sampling procedure follows the scheme below

$$\boldsymbol{\varphi}, \sigma \sim p(\boldsymbol{\varphi}, \sigma|Y) \qquad (3.21)$$
$$X \sim p(X|Y, \boldsymbol{\varphi}, \sigma) \qquad (3.22)$$
$$\boldsymbol{\theta}, \boldsymbol{\gamma} \sim p(\boldsymbol{\theta}, \boldsymbol{\gamma}|X, \boldsymbol{\varphi}) \qquad (3.23)$$

where

$$p(\boldsymbol{\varphi}, \sigma|Y) \propto p(\boldsymbol{\varphi})p(\sigma)p(Y|\boldsymbol{\varphi}, \sigma)$$

$$\propto \int p(\boldsymbol{\varphi})p(\sigma)p(X|\boldsymbol{\varphi})p(Y|X, \sigma)dX$$

$$= p(\boldsymbol{\varphi})p(\sigma)\prod_k \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{0}, C_{\boldsymbol{\varphi}_k} + \sigma_k^2 I)$$

The sampling procedure requires two MCMC samplings for (Eq. 3.21) and (Eq. 3.23) respectively, and a direct sampling from the multivariate Gaussian distribution in (Eq. 3.22). The sampling steps are repeated until convergence is reached.

Adaptive sampling scheme

One fundamental weakness of the previous sampling strategy is that the results on $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ from (Eq. 3.23) are never propagated back to (Eq. 3.21) and (Eq. 3.22), and hence have no influence on the inference of states X. To close the feedback loop, Dondelinger et al. (2013) proposed an improved sampling scheme called *adaptive gradient matching*.

Dondelinger et al. (2013) consider the joint distribution $p(\dot{X}, X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma})$, and show that its marginalization over the state derivatives $\dot{X}$ is tractable:

$$p(X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma}) = \int p(\dot{X}, X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma})d\dot{X}$$

$$= \int p(X|\boldsymbol{\varphi})p(\boldsymbol{\varphi})p(\boldsymbol{\theta})p(\boldsymbol{\gamma})p(\dot{X}|X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma})d\dot{X}$$

$$\propto \exp\left[-\frac{1}{2}\sum_k (\boldsymbol{x}_k^T C_{\boldsymbol{\varphi}_k}^{-1}\boldsymbol{x}_k + (\boldsymbol{f}_k - \boldsymbol{m}_k)^T \boldsymbol{\Lambda}_k (\boldsymbol{f}_k - \boldsymbol{m}_k))\right] \qquad (3.24)$$

The full joint distribution, after integrating over the latent state derivatives $\dot{X}$, is then given by

$$p(Y, X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \sigma, \boldsymbol{\gamma}) = p(Y|X, \sigma)p(X|\boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma})p(\boldsymbol{\varphi})p(\boldsymbol{\theta})p(\sigma)p(\boldsymbol{\gamma})$$
$$= p(Y|X, \sigma)p(X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma})p(\sigma) \qquad (3.25)$$

Based on the above joint distribution, Dondelinger et al. (2013) devised an improved Metropolis-Hastings sampling scheme, which allows $\boldsymbol{\theta}$ to exert an influence on X.

Summary

To conclude this section, given the implicit solutions from the Gaussian processes, the states and parameters of the ODEs are inferred simultaneously without explicitly solving the ODEs anymore, which has led to significant speed-up. Lastly, Calderhead et al. (2009) also discuss the handling of partial observations, i.e. when some states are not observed, by utilizing to the prior we have imposed on the states (Eq. 3.14).

3.3 Variational gradient matching with Gaussian processes

Instead of using sampling methods, Gorbach et al. (2016, 2017) proposed a solution to the previous problem based on variational inference, which has been shown, for specific types of ODEs (Eq. 3.28) as discussed below, to be much more efficient than the sampling-based solutions, and scales well to large dynamical systems such the deterministic Lorenz 96 model with 1000 states in their experiments. This section gives an outline of the method by following the description from Gorbach et al. (2017) and the method is abbreviated as VGMGP in the rest of this work.

Maximum a posteriori estimation

By combining the prior $p(\boldsymbol{\theta})$, the state posterior (Eq. 3.16), the product of experts result (Eq. 3.19), and then integrating out the latent state derivatives $\dot{X}$ as in Calderhead et al. (2009), the joint posterior $p(X, \boldsymbol{\theta}|Y, \boldsymbol{\varphi}, \sigma, \boldsymbol{\gamma})$ is given as follows:

$$p(X, \boldsymbol{\theta}|Y, \boldsymbol{\varphi}, \sigma, \boldsymbol{\gamma}) = \int p(\boldsymbol{\theta})p(X|Y, \boldsymbol{\varphi}, \sigma)p(\dot{X}|X, \boldsymbol{\theta}, \boldsymbol{\varphi}, \boldsymbol{\gamma})d\dot{X}$$
$$\propto p(\boldsymbol{\theta}) \prod_k [\mathcal{N}(\boldsymbol{x}_k|\boldsymbol{\mu}_k(\boldsymbol{y}_k), \boldsymbol{\Sigma}_k)\mathcal{N}(\boldsymbol{f}_k(X, \boldsymbol{\theta})|\boldsymbol{m}_k, \boldsymbol{\Lambda}_k^{-1})] \quad (3.26)$$

Using the inference of the parameters $\boldsymbol{\theta}$ as an example, the "best" parameters $\boldsymbol{\theta}^*$ could ideally be estimated based on the *maximum a posteriori (MAP)* of the

posterior $p(\boldsymbol{\theta}|\mathbb{Y},\boldsymbol{\varphi},\sigma,\boldsymbol{\gamma})$, which is equivalent to

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \ln p(\boldsymbol{\theta}|\mathbb{Y},\boldsymbol{\varphi},\sigma,\boldsymbol{\gamma}) \qquad (3.27)$$

where the posterior on $\boldsymbol{\theta}$ is obtained by marginalizing (Eq. 3.26) over the states $\mathbb{X}$, namely

$$p(\boldsymbol{\theta}|\mathbb{Y},\boldsymbol{\varphi},\sigma,\boldsymbol{\gamma}) = \int p(\mathbb{X},\boldsymbol{\theta}|\mathbb{Y},\boldsymbol{\varphi},\sigma,\boldsymbol{\gamma})d\mathbb{X}$$

However, the above equation is intractable due to strong non-linear couplings of the states induced by the ODEs (Eq. 3.13). As a workaround, Gorbach et al. (2017) designed a proxy distribution on the states and parameters denoted as $Q(\mathbb{X},\boldsymbol{\theta})$, and derived analytically tractable variational lower bounds based on mean-field variational inference.

Structural assumption

First of all, the ODEs primarily considered by Gorbach et al. (2017) are statewise of the following structure:

$$f_k(\boldsymbol{x}(t),\boldsymbol{\theta}) = \sum_m \theta_m \prod_{i\in\mathcal{M}_{km}} x_i(t) + \sum_j C_j \qquad (3.28)$$

for $k = 1,\ldots,K$, where $\mathcal{M}_{km}$ contains the states in the k-th equation of the ODEs that are controlled by the m-th parameter θ_m, and C_j is a term that is independent of the parameters. The first term of (Eq. 3.28) can be viewed as a linear combination of the parameters and the product of the monomials of the states. For the second term, the requirement C_j is that each state can only appear as monomial inside C_j, but C_j can contain the product of the monomials of the states.

The structural assumption imposed on the ODEs leads to an important conclusion that the following conditional distributions $p(\boldsymbol{\theta}|\mathbb{Y},\mathbb{X},\boldsymbol{\varphi},\boldsymbol{\gamma})$ and $p(\boldsymbol{x}_u|\mathbb{Y},\mathbb{X}_{/\{u\}},\boldsymbol{\varphi},\boldsymbol{\theta},\sigma,\boldsymbol{\gamma})$ for $u = 1,\ldots,K$ are Gaussian distributed, where $\mathbb{X}_{/\{u\}} = \{\boldsymbol{x}_o|o = 1,\ldots,K \text{ and } o \neq u\}$, i.e. $\mathbb{X}_{/\{u\}}$ is the set of all states except $\boldsymbol{x}_u$. But before writing out the distributions, we first need to introduce several more notations.

One implication of the above assumption is that each ODE can be expressed as a linear combination of the parameters plus a term that is independent of them. We therefore use $\boldsymbol{B}_{\theta k}$ and $\boldsymbol{b}_{\theta k}$ to denote the corresponding coefficients such that $\boldsymbol{B}_{\theta k}\boldsymbol{\theta} + \boldsymbol{b}_{\theta k} = f_k(\mathbb{X},\boldsymbol{\theta})$ for $k = 1,\ldots,K$. Similarly, due to the monomial assumption about the states, each ODE can also be transformed into a linear combination of the u-th state for $u = 1,\ldots,K$ plus an independent term. Here we adopt the terms $\boldsymbol{B}_{uk}$ and $\boldsymbol{b}_{uk}$ to denote the respective coefficients such that $\boldsymbol{B}_{uk}\boldsymbol{x}_u + \boldsymbol{b}_{uk} = f_k(\mathbb{X},\boldsymbol{\theta})$ for $k,u = 1,\ldots,k$.

With these notations, we now state the conditional distributions $p(\boldsymbol{\theta}|\mathrm{Y}, \mathrm{X}, \boldsymbol{\varphi}, \boldsymbol{\gamma})$ and $p(\boldsymbol{x}_u|\mathrm{Y}, \mathrm{X}_{/\{u\}}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\gamma})$ as follows:

$$p(\boldsymbol{\theta}|\mathrm{Y}, \mathrm{X}, \boldsymbol{\varphi}, \boldsymbol{\gamma}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{r_\theta}, \boldsymbol{\Omega_\theta}) \tag{3.29}$$

$$p(\boldsymbol{x}_u|\mathrm{Y}, \mathrm{X}_{/\{u\}}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\gamma}) = \mathcal{N}(\boldsymbol{x}_u|\boldsymbol{r}_u, \boldsymbol{\Omega}_u) \tag{3.30}$$

where

$$\boldsymbol{r_\theta} = \boldsymbol{\Omega_\theta} \sum_k \boldsymbol{B}_{\theta k}^T \boldsymbol{\Lambda}_k (\boldsymbol{m}_k - \boldsymbol{b}_{\theta k}) \tag{3.31}$$

$$\boldsymbol{\Omega_\theta}^{-1} = \sum_k \boldsymbol{B}_{\theta k}^T \boldsymbol{\Lambda}_k \boldsymbol{B}_{\theta k} \tag{3.32}$$

$$\boldsymbol{r}_u = \boldsymbol{\Omega}_u [\boldsymbol{\Sigma}_u^{-1} \boldsymbol{\mu}_u(\boldsymbol{y}_u) + \sum_k \boldsymbol{B}_{uk}^T \boldsymbol{\Lambda}_k (\boldsymbol{m}_k - \boldsymbol{b}_{uk})] \tag{3.33}$$

$$\boldsymbol{\Omega}_u^{-1} = \boldsymbol{\Sigma}_u^{-1} + \sum_k \boldsymbol{B}_{uk}^T \boldsymbol{\Lambda}_k \boldsymbol{B}_{uk} \tag{3.34}$$

The proofs are omitted here and can be found in the supporting materials of Gorbach et al. (2017). They basically utilizes the formula to calculate the product of Gaussians (Petersen and Pedersen, 2012, Section 8.1), and the well-known closure property of Gaussian distribution under linear transformations, to obtain a closed-form solution. As a side note, this conclusion will also be used later in Chapter 4 to derive another approximate inference solution based on the Laplace approximation technique.

To simplify the notations, in the following, we will switch to the canonical form of *exponential family* (Murphy, 2012, Section 9.2) to describe the distributions (Eq. 3.29) and (Eq. 3.30) as:

$$p(\boldsymbol{\theta}|\mathrm{Y}, \mathrm{X}, \boldsymbol{\varphi}, \boldsymbol{\gamma})$$
$$= h_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \exp\left(\boldsymbol{\eta}_{\boldsymbol{\theta}}(\mathrm{Y}, \mathrm{X}, \boldsymbol{\varphi}, \boldsymbol{\gamma})^T \boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{\theta}) - A_{\boldsymbol{\theta}}(\boldsymbol{\eta}_{\boldsymbol{\theta}})\right) \tag{3.35}$$
$$p(\boldsymbol{x}_u|\mathrm{Y}, \mathrm{X}_{/\{u\}}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\gamma})$$
$$= h_u(\boldsymbol{x}_u) \exp\left(\boldsymbol{\eta}_u(\mathrm{Y}, \mathrm{X}_{/\{u\}}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\gamma})^T \boldsymbol{T}_u(\boldsymbol{x}_u) - A_u(\boldsymbol{\eta}_u)\right) \tag{3.36}$$

where $\boldsymbol{\eta}_{(\cdot)}(\cdot), \boldsymbol{T}_{(\cdot)}(\cdot), h_{(\cdot)}(\cdot), A_{(\cdot)}(\cdot)$ are the *natural parameters, sufficient statistics, base measure* and *log partition function* respectively .

Variational lower bound

Introducing $\boldsymbol{\lambda}$ and $\boldsymbol{\psi}_u$ for $u = 1, \ldots, K$ as the variational parameters, the proxy distribution $Q(\mathrm{X}, \boldsymbol{\theta})$ is constrained to the family $\mathcal{Q}$ that factorizes over the parameters and the states, and each factor is in the same exponential family as its corresponding counterparts defined in (Eq. 3.35) and (Eq. 3.36) such that

$$\mathcal{Q} = \{Q(\mathrm{X}, \boldsymbol{\theta})|Q(\mathrm{X}, \boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\lambda}) \prod_u q(\boldsymbol{x}_u|\boldsymbol{\psi}_u)\} \tag{3.37}$$

where

$$q(\boldsymbol{\theta}|\boldsymbol{\lambda}) = h_{q\theta}(\boldsymbol{\theta}) \exp\left(\boldsymbol{\lambda}^T \boldsymbol{T}_{q\theta}(\boldsymbol{\theta}) - A_{q\theta}(\boldsymbol{\lambda})\right)$$
$$q(\boldsymbol{x}_u|\boldsymbol{\psi}_u) = h_{qu}(\boldsymbol{x}_u) \exp\left(\boldsymbol{\psi}_u^T \boldsymbol{T}_{qu}(\boldsymbol{\psi}_u) - A_{qu}(\boldsymbol{\psi}_u)\right)$$

Using standard mean-field variation technique, the optimal Q^* is given by

$$
\begin{aligned}
Q^* &= \underset{Q(X,\boldsymbol{\theta})\in\mathcal{Q}}{\arg\min}\, KL(Q(X,\boldsymbol{\theta})\|p(X,\boldsymbol{\theta}|Y,\boldsymbol{\varphi},\sigma,\boldsymbol{\gamma})) \\
&= \underset{Q(X,\boldsymbol{\theta})\in\mathcal{Q}}{\arg\max}\, \mathcal{L}_Q(\boldsymbol{\lambda},\boldsymbol{\psi})
\end{aligned}
\tag{3.38}
$$

where $\mathcal{L}_Q(\boldsymbol{\lambda},\boldsymbol{\psi})$ denotes the evidence lower bound.

Gorbach et al. (2017) showed that maximizing $\mathcal{L}_Q(\boldsymbol{\lambda},\boldsymbol{\psi})$ with respect to $\boldsymbol{\theta}$ is equivalent to maximizing

$$
\begin{aligned}
\mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{\lambda}) &= \mathbb{E}_Q[\ln p(\boldsymbol{\theta}|Y,X,\boldsymbol{\varphi},\boldsymbol{\gamma})] - \mathbb{E}_Q[\ln q(\boldsymbol{\theta}|\boldsymbol{\lambda})] \\
&= \mathbb{E}_Q[\boldsymbol{\eta}_{\boldsymbol{\theta}}(Y,X,\boldsymbol{\varphi},\boldsymbol{\gamma})^T \nabla_{\boldsymbol{\lambda}} A_{\boldsymbol{\theta}}(\boldsymbol{\eta}_{\boldsymbol{\theta}})] - \boldsymbol{\lambda}^T \nabla_{\boldsymbol{\lambda}} A_{q\theta}(\boldsymbol{\lambda})
\end{aligned}
\tag{3.39}
$$

and maximizing $\mathcal{L}_Q(\boldsymbol{\lambda},\boldsymbol{\psi})$ with respect to $\boldsymbol{x}_u$ is equivalent to maximizing

$$
\begin{aligned}
\mathcal{L}_{\boldsymbol{\psi}_u} &= \mathbb{E}_Q[\ln p(\boldsymbol{x}_u|Y,X_{/\{u\}},\boldsymbol{\varphi},\boldsymbol{\theta},\sigma,\boldsymbol{\gamma})] - \mathbb{E}_Q[\ln q(\boldsymbol{x}_u|\boldsymbol{\psi}_u)] \\
&= \mathbb{E}_Q[\boldsymbol{\eta}_u(Y,X_{/\{u\}},\boldsymbol{\varphi},\boldsymbol{\theta},\sigma,\boldsymbol{\gamma})^T \nabla_u A_u(\boldsymbol{\eta}_u)] - \boldsymbol{\psi}_u^T \nabla_u A_{qu}(\boldsymbol{\psi}_u)
\end{aligned}
\tag{3.40}
$$

where $\nabla_{(\cdot)}$ is the gradient operator.

Given the assumption about the conditional distributions (Eq. 3.35) and (Eq. 3.36), plus the design of the proxy distribution family (Eq. 3.37), they further showed that the lower bounds (Eq. 3.39) and (Eq. 3.40) can be optimized analytically by setting their gradients with respect to their variational parameters to zero. The optimal variational parameters are thus given by

$$\boldsymbol{\lambda}^* = \mathbb{E}_Q[\boldsymbol{\eta}_{\boldsymbol{\theta}}(Y,X,\boldsymbol{\varphi},\boldsymbol{\gamma})] \tag{3.41}$$
$$\boldsymbol{\psi}_u^* = \mathbb{E}_Q[\boldsymbol{\eta}_u(Y,X_{/\{u\}},\boldsymbol{\varphi},\boldsymbol{\theta},\sigma,\boldsymbol{\gamma})] \tag{3.42}$$

where the natural parameters are defined as $\boldsymbol{\eta}_{\boldsymbol{\theta}}(Y,X,\boldsymbol{\varphi},\boldsymbol{\gamma}) = [\boldsymbol{\Omega}_{\boldsymbol{\theta}}^{-1}\boldsymbol{r}_{\boldsymbol{\theta}}, -\frac{1}{2}\boldsymbol{\Omega}_{\boldsymbol{\theta}}^{-1}]^T$ and $\boldsymbol{\eta}_u(Y,X_{/\{u\}},\boldsymbol{\varphi},\boldsymbol{\theta},\sigma,\boldsymbol{\gamma}) = [\boldsymbol{\Omega}_u^{-1}\boldsymbol{r}_u, -\frac{1}{2}\boldsymbol{\Omega}_u^{-1}]^T$.

VGMGP algorithm

To close this section, the pseudocode of their algorithm is summarized in Algorithm 1. The algorithm first uses Gaussian process regression to obtain a smoothed estimate for the states and the distribution on the state derivatives. As noted by Gorbach et al. (2017), the Gaussian process priors come in naturally in cases where there exist unobserved states.

Algorithm 1 Pseudocode for the VGMGP algorithm.

1: **for** $u = 1, \ldots, K$ **do**
2:      Initialize $\boldsymbol{\mu}_u(\boldsymbol{y}_u)$, $\boldsymbol{\Sigma}_u$, $\boldsymbol{m}_u$ and $\boldsymbol{\Lambda}_u$ using Gaussian process regression
3: **end for**
4:
5: **while** not converged or maximum iteration not reached **do**
6:      **for** $u = 1, \ldots, K$ **do**
7:          Update the mean and variance of $\hat{q}_{\boldsymbol{\psi}_u}$ using $\hat{\boldsymbol{\theta}}^{(i)}$
8:      **end for**
9:      $\hat{\boldsymbol{\theta}}^{(i+1)} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_Q[\sum_k \ln \mathcal{N}(\boldsymbol{f}_k(\mathbf{X}, \boldsymbol{\theta}) | \boldsymbol{m}_k, \boldsymbol{\Lambda}_k^{-1})]$
10: **end while**

In the second step, the algorithm uses the variational gradient matching framework described before to maximize the lower bounds iteratively. In every iteration, the states and parameters are optimized in an *expectation-maximization (EM)* (Bishop, 2006, Section 9.4) fashion until convergence is reached or the maximum allowed number of iterations is reached.

Chapter 4

Laplace Mean-Field Approximation

One limitation of the previous VGMGP methodology is that analytical variational lower bounds are obtainable only if the structural assumption on the ODEs is satisfied. Although many dynamical systems such the Lotka-Volterra model (Section 6.2), the Lorenz 63 model (Section 6.5) and the Lorenz 96 model (Section 6.4) fulfill this requirement, it would be valuable to devise a more general solution without constraining the structure of the ODEs. Moreover, for models of biochemical or physical interactions such as the protein signaling transduction pathway model (Section 6.3), sometimes the states or the parameters need to be positive in order to give meaningful results. This positivity constraint is also not supported by VGMGP. Lastly, as can be seen from (Eq. 3.41) and (Eq. 3.42), the analytical solution requires lots of book-keeping, due to the evaluation of the expectations, which renders the implementation cumbersome and error prone.

As an extension to the previous variational approach, this chapter derives another approximation scheme based on Laplace approximation. Section 4.1 give a brief review about the general Laplace approximation technique. In Section 4.2, it is applied to the VGMGP model to derive a new solution called *Laplace mean-field (LPMF)*, which relaxes the assumption about the ODE structure. Similar to the variational approach, this solution also transforms the original inference problem into an optimization problem. However, due to the relaxation on the ODE structure, the optimization objectives can in general no longer be optimized in closed-form, and hence, numerical solutions have to be used. As will be shown in Section 4.3, the gradients and even the Hessians of the state objectives can be computed efficiently, which allows us to rely on second-order optimization techniques and enables the algorithm to scale to large-scale dynamical systems. Viewing the LPMF optimization objectives as risk functions, we can further use the reparameterization trick to enforce positivity constraints on the states and the ODE parameters, which will be discussed in Section 4.4. The strengths and

weaknesses of the LPMF solution are examined empirically and discussed in Chapter 6.

4.1 Laplace approximation

This section reviews the *Laplace approximation* technique in the context of approximating an unknown multivariate probability distribution based on (Bishop, 2006, Section 4.4) and (MacKay, 2003, Section 27).

Suppose we are interested in the following probability distribution

$$p(x) = \frac{\widetilde{p}(x)}{Z} \tag{4.1}$$

where $x \in \mathbb{R}^D$, $\widetilde{p}(x)$ is known and $Z = \int \widetilde{p}(x)dx$ is the normalization constant assumed to be intractable. The Laplace approximation of p is a Gaussian distribution q such that its mean is centered on a mode of the p (MacKay, 2003).

Assuming that p has a peak at the point x_0, then the gradient $\nabla p(x_0) = 0$, or equivalently, $\nabla \widetilde{p}(x_0) = 0$. The second-order *Taylor expansion* of $\ln \widetilde{p}(x)$ around x_0 is given by

$$\ln \widetilde{p}(x) \approx \ln \widetilde{p}(x_0) + (x - x_0)\nabla \widetilde{p}(x_0) - \frac{1}{2}(x - x_0)^T \widetilde{H}(x - x_0)$$
$$= \ln \widetilde{p}(x_0) - \frac{1}{2}(x - x_0)^T \widetilde{H}(x - x_0) \tag{4.2}$$

where $\widetilde{H}$ is the negative of the Hessian matrix H at x_0 such that

$$\widetilde{H}_{ij} = -H_{ij} = -\frac{\partial^2}{\partial x_i \partial x_j} \ln \widetilde{p}|_{x=x_0} \tag{4.3}$$

or

$$\widetilde{H} = -H = -\nabla\nabla \ln \widetilde{p}(x_0) \tag{4.4}$$

Exponentiating both sides of (Eq. 4.2), we have

$$\widetilde{p}(x) \approx \widetilde{p}(x_0) \exp\left[-\frac{1}{2}(x - x_0)^T \widetilde{H}(x - x_0)\right] \tag{4.5}$$

which is of quadratic form and can be normalized by inspection to obtain the following Gaussian distribution:

$$q(x) = \frac{|\widetilde{H}|^{\frac{1}{2}}}{(2\pi)^{\frac{D}{2}}} \exp\left[-\frac{1}{2}(x - x_0)^T \widetilde{H}(x - x_0)\right]$$
$$= \mathcal{N}(x|x_0, \widetilde{H}^{-1}) \tag{4.6}$$

4.2 Laplace mean-field approximation

As discussed in Section 3.3, the conditional distributions $p(\boldsymbol{\theta}|Y, X, \boldsymbol{\varphi}, \boldsymbol{\gamma})$ (Eq. 3.30) and $p(\boldsymbol{x}_u|Y, X_{/\{u\}}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\gamma})$ (Eq. 3.29) are both Gaussians provided that the ODEs fulfill the structural assumption described by (Eq. 3.28). If we relax the constraint on the structure of the ODEs, the distributions $p(\boldsymbol{\theta}|Y, X, \boldsymbol{\varphi}, \boldsymbol{\gamma})$ and $p(\boldsymbol{x}_u|Y, X_{/\{u\}}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\gamma})$ are no longer Gaussians and cannot be normalized in closed-form anymore.

From the gradient matching model, we have

$$
\begin{aligned}
p(\boldsymbol{\theta}|Y, X, \boldsymbol{\varphi}, \boldsymbol{\gamma}) &\stackrel{(a)}{=} p(\boldsymbol{\theta}|X, \boldsymbol{\varphi}, \boldsymbol{\gamma}) \\
&= \int p(\boldsymbol{\theta}) p(\dot{X}|X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma}) d\dot{X} \\
&\propto \prod_k \mathcal{N}(\boldsymbol{f}_k(X, \boldsymbol{\theta})|\boldsymbol{m}_k, \boldsymbol{\Lambda}_k^{-1})
\end{aligned}
\tag{4.7}
$$

where (a) holds since $\boldsymbol{\theta}$ depends indirectly on Y through X (Gorbach et al., 2017), and $p(\dot{X}|X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma})$ is the product of experts result (Eq. 3.19).

Similarly, for each state $\boldsymbol{x}_u$ with $u = 1, \ldots, K$, we have

$$
\begin{aligned}
p(\boldsymbol{x}_u|Y, X_{/\{u\}}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\gamma}) &= \int p(\boldsymbol{x}_u|Y, X_{/\{u\}}, \boldsymbol{\varphi}, \boldsymbol{\sigma}) p(\dot{X}|\boldsymbol{x}_u, X_{/\{u\}}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma}) d\dot{X} \\
&\stackrel{(b)}{=} \int p(\boldsymbol{x}_u|\boldsymbol{y}_u, \boldsymbol{\varphi}_k, \sigma_k) p(\dot{X}|X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma}) d\dot{X} \\
&\propto \mathcal{N}(\boldsymbol{x}_u|\boldsymbol{\mu}_u(\boldsymbol{y}_u), \boldsymbol{\Sigma}_u) \prod_k \mathcal{N}(\boldsymbol{f}_k(X, \boldsymbol{\theta})|\boldsymbol{m}_k, \boldsymbol{\Lambda}_k^{-1})
\end{aligned}
\tag{4.8}
$$

where (b) holds because $p(\boldsymbol{x}_u|Y, X_{/\{u\}}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$ depends only on $\boldsymbol{y}_u$ as the consequence of the independent Gaussian process prior assumption on states (Eq. 3.14), and $p(\dot{X}|\boldsymbol{x}_u, X_{/\{u\}}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma})$ is equivalent to $p(\dot{X}|X, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\gamma})$.

Cost functions

If we follow the same factorization assumption over the states and the parameters as Gorbach et al. (2017), and require each component of the proxy distribution $Q(X, \boldsymbol{\theta})$ to be Gaussian, then $Q(X, \boldsymbol{\theta})$ is of the following form

$$
\begin{aligned}
Q(X, \boldsymbol{\theta}) &= q(\boldsymbol{\theta}|\boldsymbol{\eta}_{\boldsymbol{\theta}}, \boldsymbol{\Xi}_{\boldsymbol{\theta}}) \prod_u q(\boldsymbol{x}_u|\boldsymbol{\eta}_{\boldsymbol{x}_u}, \boldsymbol{\Xi}_{\boldsymbol{x}_u}) \\
&= \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\eta}_{\boldsymbol{\theta}}, \boldsymbol{\Xi}_{\boldsymbol{\theta}}) \prod_u \mathcal{N}(\boldsymbol{x}_u|\boldsymbol{\eta}_{\boldsymbol{x}_u}, \boldsymbol{\Xi}_{\boldsymbol{x}_u})
\end{aligned}
\tag{4.9}
$$

Based on (Eq. 4.7) and the Laplace approximation technique reviewed before, the mean of the Gaussian proxy $q(\boldsymbol{\theta}|\boldsymbol{\eta}_{\boldsymbol{\theta}}, \boldsymbol{\Xi}_{\boldsymbol{\theta}})$ can be found by

$$
\begin{aligned}
\boldsymbol{\eta}_{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} \ln \prod_k \mathcal{N}(f_k(\mathrm{X}, \boldsymbol{\theta})|\boldsymbol{m}_k, \boldsymbol{\Lambda}_k^{-1}) \\
&= \arg\max_{\boldsymbol{\theta}} \sum_k \ln \mathcal{N}(f_k(\mathrm{X}, \boldsymbol{\theta})|\boldsymbol{m}_k, \boldsymbol{\Lambda}_k^{-1}) \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \sum_k (f_k - \boldsymbol{m}_k)^T \boldsymbol{\Lambda}_k (f_k - \boldsymbol{m}_k) \\
&= \arg\min_{\boldsymbol{\theta}} cost_{\boldsymbol{\theta}}(\mathrm{X}, \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\Lambda})
\end{aligned}
\tag{4.10}
$$

where $f_k = f_k(\mathrm{X}, \boldsymbol{\theta})$, $\boldsymbol{m} = [\boldsymbol{m}_1, \ldots, \boldsymbol{m}_K]$ and $\boldsymbol{\Lambda} = [\boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_K]$. The precision matrix $\boldsymbol{\Xi}_{\boldsymbol{\theta}}^{-1}$ is then the Hessian of the objective function (Eq. 4.10) evaluated at the optimal point $\boldsymbol{\eta}_{\boldsymbol{\theta}}$

$$
\boldsymbol{\Xi}_{\boldsymbol{\theta}}^{-1} = \nabla\nabla cost_{\boldsymbol{\theta}}|_{\boldsymbol{\theta}=\boldsymbol{\eta}_{\boldsymbol{\theta}}}
\tag{4.11}
$$

Similarly, the mean of the Gaussian proxy $q(\boldsymbol{x}_u|\boldsymbol{\eta}_{\boldsymbol{x}_u}, \boldsymbol{\Xi}_{\boldsymbol{x}_u})$ for $u = 1, \ldots, K$ can be found using (Eq. 4.8) as

$$
\begin{aligned}
\boldsymbol{\eta}_{\boldsymbol{x}_u} &= \arg\max_{\boldsymbol{x}_u} \ln \left[\mathcal{N}(\boldsymbol{x}_u|\boldsymbol{\mu}_u(\boldsymbol{y}_u), \boldsymbol{\Sigma}_u) \prod_k \mathcal{N}(f_k(\mathrm{X}, \boldsymbol{\theta})|\boldsymbol{m}_k, \boldsymbol{\Lambda}_k^{-1}) \right] \\
&= \arg\max_{\boldsymbol{x}_u} \left[\ln \mathcal{N}(\boldsymbol{x}_u|\boldsymbol{\mu}_u(\boldsymbol{y}_u), \boldsymbol{\Sigma}_u) + \sum_k \ln \mathcal{N}(f_k(\mathrm{X}, \boldsymbol{\theta})|\boldsymbol{m}_k, \boldsymbol{\Lambda}_k^{-1}) \right] \\
&= \arg\min_{\boldsymbol{x}_u} \frac{1}{2} \left[(\boldsymbol{x}_u - \boldsymbol{\mu}_u(\boldsymbol{y}_u))^T \boldsymbol{\Sigma}_u^{-1}(\boldsymbol{x}_u - \boldsymbol{\mu}_u(\boldsymbol{y}_u)) + \sum_k (f_k - \boldsymbol{m}_k)^T \boldsymbol{\Lambda}_k(f_k - \boldsymbol{m}_k) \right] \\
&= \arg\min_{\boldsymbol{x}_u} cost_{\boldsymbol{x}_u}(\boldsymbol{x}_u, \mathrm{X}_{/\{u\}}, \boldsymbol{\theta}, \boldsymbol{\mu}_u(\boldsymbol{y}_u), \boldsymbol{\Sigma}_u, \boldsymbol{m}, \boldsymbol{\Lambda})
\end{aligned}
\tag{4.12}
$$

The corresponding precision matrix $\boldsymbol{\Xi}_{\boldsymbol{x}_u}^{-1}$ is given by

$$
\boldsymbol{\Xi}_{\boldsymbol{x}_u}^{-1} = \nabla\nabla cost_{\boldsymbol{x}_u}|_{\boldsymbol{x}_u=\boldsymbol{\eta}_{\boldsymbol{x}_u}}
\tag{4.13}
$$

LPMF algorithm

To conclude, given the initialization from the Gaussian processes, the optimization step (Eq. 4.10) first updates the estimation of the parameters, and then for each state, the optimization procedure (Eq. 4.19) improves state estimation. These two steps are executed in turn iteratively until convergence of the estimation or the maximum allowed number of iterations is reached. Lastly, the covariance matrices of the estimation results are calculated using (Eq. 4.11) and (Eq. 4.13). The pseudocode for the LPMF algorithm is given in Algorithm 2.

Note that when the ODEs satisfy the structural assumption about the parameters $\boldsymbol{\theta}$, closed-form solutions for $\boldsymbol{\eta}_{\boldsymbol{\theta}}$ and $\boldsymbol{\Xi}_{\boldsymbol{\theta}}$ can be obtained using (Eq.

Algorithm 2 Pseudocode for the LPMF algorithm.

1: **for** $u = 1, \ldots, K$ **do**
2:     Initialize $\mu_u(y_u)$, Σ_u, m_u and Λ_u using Gaussian process regression
3:     $\eta_{x_u} = \mu_u(y_u)$
4: **end for**
5:
6: **while** not converged or maximum iteration not reached **do**
7:     $\eta_\theta = \arg\min_\theta cost_\theta(\eta_X, \theta, m, \Lambda)$
8:     **for** $u = 1, \ldots, K$ **do**
9:         $\eta_{x_u} = \arg\min_{x_u} cost_{x_u}(x_u, \eta_{X_{/\{u\}}}, \eta_\theta, \mu_u(y_u), \Sigma_u, m, \Lambda)$
10:     **end for**
11: **end while**
12:
13: $\Xi_\theta^{-1} = \nabla\nabla cost_\theta(\eta_X, \eta_\theta, m, \Lambda)$
14: **for** $u = 1, \ldots, K$ **do**
15:     $\Xi_{x_u}^{-1} = \nabla\nabla cost_{x_u}(\eta_{x_u}, \eta_{X_{/\{u\}}}, \eta_\theta, \mu_u(y_u), \Sigma_u, m, \Lambda)$
16: **end for**

3.31) and (Eq. 3.32) respectively, which requires rewriting the ODEs as linear combination of the parameters plus a term that is independent of them. Similarly, analytical solutions for η_{x_u} and Ξ_{x_u} can be derived from (Eq. 3.33) and (Eq. 3.34) for $u = 1, \ldots, K$ when the structural assumption about the states is fulfilled. This also requires rewriting of the ODEs as a linear combination of the states plus an independent term, which is more cumbersome than the previous case for dynamical systems with large number of states.

4.3   Derivation for the gradients and Hessians

This section derives the gradient and Hessian for the cost function $cost_{x_u}$ (Eq. 4.19) and shows that they can be evaluated efficiently.

Recall that the cost function $cost_{x_u}$ for state u is given by

$$cost_{x_u} = \frac{1}{2}[(x_u - \mu_u(y_u))^T \Sigma_u^{-1}(x_u - \mu_u(y_u)) + \sum_k (f_k - m_k)^T \Lambda_k (f_k - m_k)]$$

We are interested in the gradient and Hessian of $cost_{x_u}$ w.r.t state x_u, i.e. $\nabla_{x_u} cost_{x_u}$ and $\nabla\nabla_{x_u} cost_{x_u}$. Since differentiation is a linear operation, we can look at the gradient and Hessian of each component of the cost function separately.

Using matrix derivative and the fact that Σ_u^{-1} is symmetric, the gradient and

Hessian of the first term in $cost_{x_u}$ are

$$\nabla_{x_u} \frac{1}{2}(x_u - \mu_u(y_u))^T \Sigma_u^{-1}(x_u - \mu_u(y_u)) = \Sigma_u^{-1} x_u \tag{4.14}$$

and

$$\nabla \nabla_{x_u} \frac{1}{2}(x_u - \mu_u(y_u))^T \Sigma_u^{-1}(x_u - \mu_u(y_u)) = \Sigma_u^{-1} \tag{4.15}$$

Using the *chain rule* and the fact that Λ_k is symmetric, for each component from the second term in $cost_{x_u}$, the gradient is given by

$$
\nabla_{x_u} \frac{1}{2}(f_k - m_k)^T \Lambda_k(f_k - m_k)
$$

$$
= \begin{bmatrix} \frac{\partial(f_k - m_k)_1}{\partial x_u(t_1)} & \cdots & \frac{\partial(f_k - m_k)_N}{\partial x_u(t_1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial(f_k - m_k)_1}{\partial x_u(t_N)} & \cdots & \frac{\partial(f_k - m_k)_N}{\partial x_u(t_N)} \end{bmatrix} \Lambda_k(f_k - m_k)
$$

$$
= \begin{bmatrix} \frac{\partial(f_k)_1}{\partial x_u(t_1)} & \cdots & \frac{\partial(f_k)_N}{\partial x_u(t_1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial(f_k)_1}{\partial x_u(t_N)} & \cdots & \frac{\partial(f_k)_N}{\partial x_u(t_N)} \end{bmatrix} \Lambda_k(f_k - m_k)
$$

$$
- \begin{bmatrix} \frac{\partial(m_k)_1}{\partial x_u(t_1)} & \cdots & \frac{\partial(m_k)_N}{\partial x_u(t_1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial(m_k)_1}{\partial x_u(t_N)} & \cdots & \frac{\partial(m_k)_N}{\partial x_u(t_N)} \end{bmatrix} \Lambda_k(f_k - m_k) \tag{4.16}
$$

Since the states do not appear inside the vector field across time points, the first matrix above is necessarily a diagonal matrix. If the f_k is independent of x_u, then this matrix is a zero matrix. Since $m_k = C''_{\varphi_k} C_{\varphi_k}^{-1} x_k$ is a linear combination of x_k, the second matrix is the transpose of $C''_{\varphi_k} C_{\varphi_k}^{-1}$ if $k = u$ and is a zero matrix otherwise.

The Hessian of $\frac{1}{2}(f_k - m_k)^T \Lambda_k(f_k - m_k)$ w.r.t. x_u also exhibits such sparse structure. In brief, the (i, j)-th entry of the Hessian is given by

$$
\frac{\partial^2 \frac{1}{2}(f_k - m_k)^T \Lambda_k(f_k - m_k)}{\partial x_u(t_i)\partial x_u(t_j)}
$$

$$
= \begin{bmatrix} \frac{\partial^2(f_k - m_k)_1}{\partial x_u(t_i)\partial x_u(t_j)} & \cdots & \frac{\partial^2(f_k - m_k)_N}{\partial x_u(t_i)\partial x_u(t_j)} \end{bmatrix} \Lambda_k(f_k - m_k)
$$

$$
+ \begin{bmatrix} \frac{\partial(f_k - m_k)_1}{\partial x_u(t_j)} & \cdots & \frac{\partial(f_k - m_k)_N}{\partial x_u(t_j)} \end{bmatrix} \Lambda_k \begin{bmatrix} \frac{\partial(f_k - m_k)_1}{\partial x_u(t_i)} \\ \vdots \\ \frac{\partial(f_k - m_k)_N}{\partial x_u(t_i)} \end{bmatrix} \tag{4.17}
$$

The overall gradients and Hessian are just the sums of the respective components discussed above. With proper handling of multiplication involving diagonal matrices and caching of constant terms, the gradient and Hessian of of the cost function $cost_{x_u}$ can be very efficiently calculated, as will be shown empirically in Chapter 6.

Although the gradient and Hessian of $cost_\theta$ (Eq. 4.10) w.r.t. θ do not have such structure, they can still be evaluated relatively efficiently since the number of parameters is usually small in comparison to the number of states. In this work, both symbolic tools and machine learning libraries supporting auto-differentiation are employed to calculate the gradient and Hessian of $cost_\theta$.

4.4 Positivity constraints

An advantage of the LPMF solution is that we can treat the optimization objectives $cost_\theta$ and $cost_{x_u}$ for $u = 1, \ldots, K$ as risk functions, and hence the name "cost" is used in the notations. This allows us to apply the following reparameterization trick to enforce a positivity constraint on the parameters and the states.

From Section 4.2, the optimal estimation for the parameters and states is given by

$$\theta^* = \arg \min_{\theta} cost_\theta(X, \theta, m, \Lambda) \tag{4.18}$$

$$x_u^* = \arg \min_{x_u} cost_{x_u}(x_u, X_{/\{u\}}, \theta, \mu_u(y_u), \Sigma_u, m, \Lambda) \tag{4.19}$$

for $u = 1, \ldots, K$. Suppose we desire the parameters to be positive values. Instead of inferring θ directly, the reparameterization trick transforms the cost function $cost_\theta$ into a new cost function $cost_{\widetilde{\theta}}$, where we define $\theta = [\theta_1, \ldots, \theta_M]^\top = [e^{\widetilde{\theta}_1}, \ldots, e^{\widetilde{\theta}_M}]^\top$. Because the exponential function is monotonic, we can therefore first find $\widetilde{\theta}^*$ that minimizes the new cost

$$\widetilde{\theta}^* = \arg \min_{\widetilde{\theta}} cost_\theta(X, e^{\widetilde{\theta}}, m, \Lambda)$$

$$= \arg \min_{\widetilde{\theta}} \sum_k \ln \mathcal{N}(f_k(X, e^{\widetilde{\theta}}) | m_k, \Lambda_k^{-1})) \tag{4.20}$$

and then exponentiate it element-wise to obtain an estimation for $\theta^* = [e^{\widetilde{\theta}_1^*}, \ldots, e^{\widetilde{\theta}_M^*}]^\top$, which would correspond to the configuration that minimizes the original $cost_\theta$. Since $e^r > 0$ for any $r \in \mathbb{R}$, we essentially restrict θ^* to only contain positive values.

Analogously, the positivity constraint on states can be achieved by transforming the cost function $cost_{x_u}$ to its equivalent $cost_{\widetilde{x}_u}$, where we define

$x_u = [x_u(t_1), \ldots, x_u(t_N)]^\top = [e^{\widetilde{x}_u(t_1)}, \ldots, e^{\widetilde{x}_u(t_N)}]^\top$ for $u = 1, \ldots, K$. Using the chain rule, we have for $u = 1, \ldots, K$ and $n = 1, \ldots, N$ the following:

$$
\begin{aligned}
\frac{d\widetilde{x}_u(t_n)}{dt} &= \frac{d \ln x_u(t_n)}{dt} \\
&= \frac{1}{x_u(t_n)} \frac{dx_u(t_n)}{dt} \\
&= \frac{f_u(e^{\widetilde{x}(t_n)}, \boldsymbol{\theta})}{e^{\widetilde{x}(t_n)}}
\end{aligned}
\tag{4.21}
$$

Note that the application of the positivity constraint on states is flexible in the sense that not all the states are required to be constrained at the same time. Also, the constraint can theoretically be applied to both the parameters and the states at the same time.

To distinguish from the unconstrained Laplace mean-field approximation, we use the term LPMP-POS to refer to this extension. The pseudocode for this extension is essentially the same as Algorithm 2 except for the replacement of the optimization functions and variables with the extra step to transform the inference result back to the original variables at the end.

One caveat for this approach is that the covariance matrix calculated for the transformed variables cannot be transformed back to the original variables easily. Also, interpretability from a probabilistic point of view is lost since exponentiation is a non-linear operation and rigorous treatment of the change of random variables would require the evaluation of the relevant Jacobian determinant, which is computationally very expensive.

Chapter 5

Extension to Random Dynamical Systems

As another important family of differential equations, *random ordinary differential equations (RODEs)* are closely related to both ODEs and SDEs. A system of RODEs is simply a set of ODEs with a stochastic process in its vector field functions (Kloeden and Jentzen, 2007), while an SDE system can be analyzed using its RODE counterpart (Sussmann, 1978; Imkeller and Schmalfuss, 2001). Since RODEs are pathwise ODEs, the Laplace mean-field approximation described in Chapter 4 can then be used to infer the states and parameters of the RODEs, or equivalently, the corresponding SDEs.

This chapter is organized as follows. Section 5.1 gives a brief introduction to RODEs. In Section 5.3, the Laplace mean-field approximation technique is applied to RODEs to devise an *ensemble*-like (Murphy, 2012, Section 16.6) solution to infer the states and parameters for diffusion processes. We demonstrate the performance and accuracy of the solution empirically by comparing with other state-of-art techniques in Chapter 6.

5.1 Random ordinary differential equations

Adopting the definition from Kloeden and Jentzen (2007), RODEs are simply ODEs with a stochastic process in their vector fields. Let $f : \mathbb{R}^K \times \mathbb{R}^W \mapsto \mathbb{R}^K$ be a continuous function, and $(\zeta_t)_{t \in [0,T]}$ be an $\mathbb{R}^W$ valued stochastic process with continuous sample paths defined on the complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$. For all $\omega \in \Omega$, a K-dimensional RODE defined as

$$\frac{dx(t)}{dt} = f(x(t), \zeta_t(\omega)) \tag{5.1}$$

is a *non-autonomous* ODE system

$$\dot{x}(t) = \frac{dx(t)}{dt} = F_\omega(x, t) = f(x(t), \omega_t) \tag{5.2}$$

An example (Grüne and Kloeden, 2001) of a scalar RODE with additive noise is given by

$$\frac{dx(t)}{dt} = -x + \cos\left(W_t(\omega)\right) \tag{5.3}$$

where W_t is a one-dimensional Wiener process. A RODE example with multiplicative noise can be defined similarly but is not considered in this work.

Following Kloeden and Jentzen (2007), to ensure the existence of a unique solution for the initial value problem defined in Section 2.1 on the finite time interval $[0, T]$, we typically assume that f is arbitrarily smooth, i.e. it is infinitely differentiable in its variables, and thus is locally *Lipschitz* in x. Since the stochastic process is usually only *Hölder continuous* in time, the vector fields of the non-autonomous ODEs $F_\omega(x, t)$ are therefore continuous but not differentiable in time for every fixed realization of $\omega \in \Omega$.

Because for every fixed realization of $\omega \in \Omega$, the RODEs (Eq. 5.1) turn into a deterministic ODE system (Eq. 5.2), one approach to solving the RODEs is to use sampling methods to first obtain many sample paths, and then solving each sample path deterministically. In order to cover the statistics of the solution, a massive number of ODEs must be solved efficiently. A high performance related study was conducted by Riesinger et al. (2016), where the sample paths are solved in parallel on a modern GPU cluster.

5.2 Doss-Sussmann/Imkeller-Schmalfuss correspondence

Since a system of RODEs can be analyzed pathwise using deterministic calculus, it offers an opportunity to study its related SDEs as discussed below.

First of all, it has been shown by Jentzen and Kloeden (2011) that any RODE system with a Wiener process can be expressed as its equivalent SDE system. Using the scalar RODE in (Eq. 5.3) as an example, its SDE formulation is described as

$$d\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} -x_t + \cos\left(y_t\right) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} dW_t \tag{5.4}$$

Similarly, any finite dimensional SDE system can be transformed into its equivalent ODEs by utilizing the *Doss-Sussmann/Imkeller-Schalfuss correspondence* (Sussmann, 1978; Imkeller and Schmalfuss, 2001). Specifically, for SDE models with additive noise, the statement from (Jentzen and Kloeden, 2011, Chapter 2) is stated as the following proposition.

Proposition 5.1 *Any finite dimensional SDE system can be transformed into an equivalent RODE system and vice versa as follows:*

$$dx(t) = f(x(t), \theta)dt + dW_t \iff \frac{dz(t)}{dt} = f(z(t) + O_t, \theta) + O_t \qquad (5.5)$$

where $z(t) = x(t) - O_t$ *and* O_t *is a stationary stochastic* Ornstein-Uhlenbeck *process defined as*

$$dO_t = -O_t dt + dW_t \qquad (5.6)$$

5.3 Laplace mean-field for random dynamical systems

As discussed previously, although RODE sample paths can be analyzed using deterministic calculus, the existence of the stochastic process causes traditional numerical schemes such as the *Euler* and the *Runge-Kutta* methods (Butcher, 2016) to fail to achieve their usual order of convergence when applied to RODEs (Grüne and Kloeden, 2001). In the past, improved numerical solutions such as the integral versions of the implicit Taylor-like expansions (Kloeden and Jentzen, 2007) have been proposed to achieve better result.

On the other hand, since the solution paths of the RODEs are once differentiable, the gradient matching model can be ideally applied to the RODEs. Moreover, the computational efficiency of the LPMF method allows a large number of RODE sample paths to be solved simultaneously. Lastly, by using the Doss-Sussmann/Imkeller-Schalfuss correspondence described before, we derive the following ensemble gradient matching algorithm, denoted as LPMF-SDE, to infer the states and the parameters of the SDEs without requiring any stochastic calculus.

Algorithm 3 Pseudocode for the LPMF-SDE algorithm.

1: Transform the SDEs into RODEs using (Eq. 5.5).
2: Generate N_{paths} RODEs sample paths using each time an independently generated Ornstein-Uhlenbeck process sample path.
3: **for** $i = 1, \ldots, N_{paths}$ **do**
4:     Estimate the states and parameters using Algorithm 2.
5: **end for**
6: Average the estimation results from all the sample paths.

Since the experiments are conducted with the Lorenz 63 and the Lorenz 96 models, the estimation of the parameters are carried out by extending the

closed-forms solutions from (Eq. 3.31) and (Eq. 3.32) as follows:

$$\eta_{\theta} = \Omega_{\theta} \sum_k B_{\theta k}^T \Lambda_k (m_k - b_{\theta k} - O_k) \tag{5.7}$$

$$\Xi_{\theta}^{-1} = \sum_k B_{\theta k}^T \Lambda_k B_{\theta k} \tag{5.8}$$

where

$$m_k = C_{\varphi_k}'' C_{\varphi_k}^{-1} z_k \tag{5.9}$$

$$\Lambda_k^{-1} = C_{\varphi_k}'' - {}'C_{\varphi_k} C_{\varphi_k}^{-1} C_{\varphi_k}' + \gamma_k I \tag{5.10}$$

Similar to the notations introduced in Section 2.1, $z_u \in \mathbb{R}^N$ and $Z \in \mathbb{R}^{K \times N}$ refer to the states of the RODE sample path, while O refer to the states of the Ornstein-Uhlenbeck process. Note that the rewriting of the vector field as a linear combination of the parameters θ and an extra term should satisfy

$$B_{\theta k}(Z + O)\theta + b_{\theta k}(Z + O) + O = f_k(Z + O, \theta) + O \tag{5.11}$$

for $k = 1, \ldots, K$.

Due to the complexity of expressing as a linear combination of the states and an extra term, the states of the RODE sample path are estimated using gradient-descent by minimizing the adapted cost function (Eq. 4.19) as follows:

$$cost_{z_u} = \ln\left[\mathcal{N}(z_u | \mu(y_u), \Sigma_u) \prod_k \mathcal{N}(f_k(Z + O, \theta) + O | m_k, \Lambda_k^{-1}))\right] \tag{5.12}$$

for $u = 1, \ldots, K$, where

$$\mu_u(y_u) = C_{\varphi_u}(C_{\varphi_u} + \sigma_u^2 I)^{-1} y_u \tag{5.13}$$

$$\Sigma_u = \sigma_u^2 C_{\varphi_u}(C_{\varphi_u} + \sigma_u^2 I)^{-1} \tag{5.14}$$

Chapter 6

Experiments

This chapter examines the estimation accuracy, runtime performance and scalability of the general LPMF method and its extensions introduced in Chapter 4 and Chapter 5 by comparing with the other state-of-art inference techniques empirically. In Section 6.1, the implementation of the algorithms is discussed. Then four dynamical systems with different dimensionality and complexity are used for the experiments. Section 6.2 and Section 6.3 consider the state and parameter estimation for deterministic dynamical systems, while Section 6.4 and Section 6.5 consider two random dynamical systems.

6.1 Implementation

The source code of this work is implemented from the ground up using the general purpose programming language Python 3[1]. The MATLAB[2] code for the VGMGP algorithm from Gorbach et al. (2017) is used as the blueprint during implementation. The advantages the Python implementation includes

- The native support of object-oriented programming makes the application interface clean and extensible. Specifically, the `Dynamical` and `Kernel` classes are implemented as abstract classes so that new dynamical systems and new kernels can be prototyped quickly.

- The language is none proprietary and many Python machine learning libraries are well supported in the open-source community and are widely used in enterprise level applications.

Several important open-source packages have made this Python solution possible:

---

[1]https://www.python.org/
[2]https://www.mathworks.com/products/matlab.html

- The SciPy[3] package provides a powerful optimization module and other I/O utilities to read and write files in MATLAB format.

- The NumPy[4] package is the backbone for linear algebra operations and random variable generation.

- The SymPy[5] package is used for symbolic mathematics when implementing the general interface for dynamical systems and kernel functions. It is also used to calculate the gradients and Hessians of the cost functions for part of the solution.

- The matplotlib[6] package helps to produce publication quality plotting.

- The Jupyter Notebook[7] is used as the GUI interface to combine code, visualizations and documentation in a sharable format.

- The sdeint[8] package provides the numerical methods to solve SDEs.

- The TensorFlow[9] package is a machine learning library popular among the deep learning community, which provides auto-differentiation support for part of the solution.

6.2 Lotka-Volterra model

The first deterministic dynamical system examined in this chapter is the *Lotka-Volterra* model (Lotka, 1932), which is frequently used in ecology to describe the interaction between the prey species and the predator species over time. The model consists of two first-order, nonlinear differential equations where the states $x(t), y(t) \in \mathbb{R}_{\geqslant 0}$ are the populations of the prey and the predator respectively at time point t. The ODEs of the model are given by

$$\dot{x}(t) = \alpha x(t) - \beta x(t)y(t)$$
$$\dot{y}(t) = \delta x(t)y(t) - \gamma y(t) \tag{6.1}$$

where $\alpha, \beta, \delta, \gamma \in \mathbb{R}^+$ are the parameters controlling the dynamics.

Experimental setup

Table 6.1 shows the setup for the experiment. The experiment is repeated 10 times with each time an independently collected observation set. Since

---

[3]https://www.scipy.org/
[4]http://www.numpy.org/
[5]http://www.sympy.org/en/index.html
[6]http://matplotlib.org/
[7]http://jupyter.org/
[8]https://github.com/mattja/sdeint
[9]https://www.tensorflow.org/

Table 6.1: Experimental setup for the Lotka-Volterra model. The system dimension is denoted by K and the number of observable dimensions is K_{obs}. Based on the parameter values α, β, δ, and γ, the ODEs are integrated from time t_0 to t_T with a step size of δt. The observation noise variance σ_k^2 is assumed to be the identical for each state. For each time unit, $freq_{obs}$ denotes the number of observations to be collected, which are equally distributed over the time line.

K	K_{obs}	t_0	t_T	δt	$\alpha, \beta, \delta, \gamma$	σ_k^2	$freq_{obs}$
2	2	0	2	0.01	2, 1, 1, 4	0.1	10
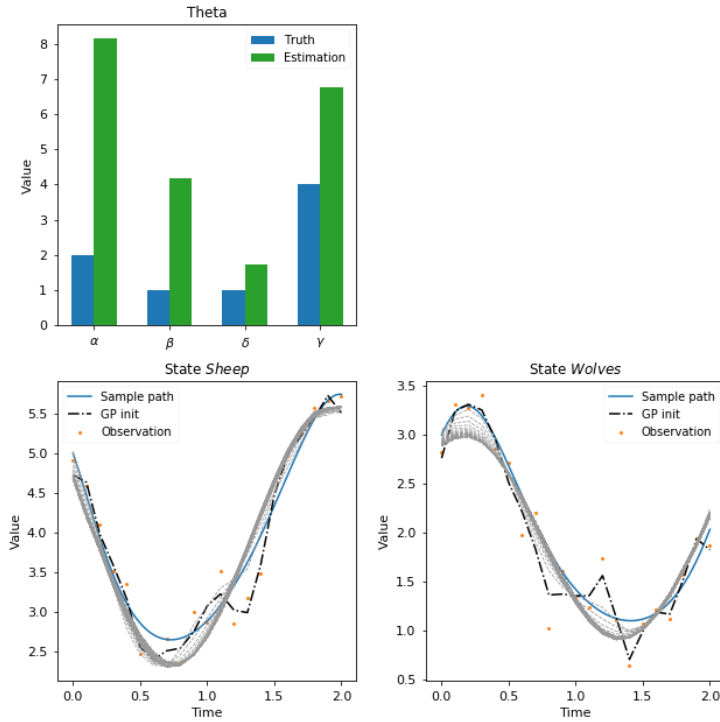
Figure 6.1: Inference for the Lotka-Volterra model fails when both the states and parameters are constrained to be positive.

the LPMF method is derived from the VGMGP method, in the following, we compare the results using both methods. The LPMF method is run first without any positivity constraint and then it is run again with positivity constraint on the parameters, which are referred to as LPMF and LPMP-POS respectively. It would be interesting to constrain both the states and the parameters to be positive, but the inference fails as shown in Figure 6.1. The reason for that in unclear yet due to time constraints and requires further investigation. In order to provide a fair comparison in terms of runtime, both methods are deployed on a desktop with an Intel i5 quad-core CPU and 16 GB of memory.
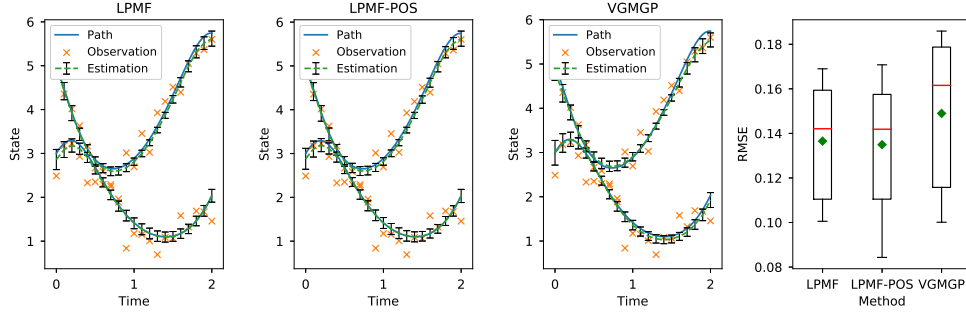
Figure 6.2: State estimation results for the Lotka-Volterra model. The left plot shows the result using LPMF; the left middle plot shows the result using LPMP-POS; the right middle plot shows the result using VGMGP; the right plot summarizes the RMSE after 10 independent runs. The error bars in the first three plots indicate one standard deviation.

State estimation

Figure 6.2 shows the results after the 10 independent runs. For illustration purposes, the observations from one run is also plotted to indicate the noise level. The dotted green lines are obtained by averaging the means of the state estimation for the 10 runs, while the error bars indicate one standard deviation of the means. The figure shows that the state estimation results are very close to each other and is almost identical to the ground truth.

To quantify the accuracy, the root mean square error (RMSE) is used and is defined as follows:

$$RMSE = \frac{1}{K} \sum_k \sqrt{\frac{1}{N} \sum_{n=1}^{N} (\hat{x}_k(t_n) - x_k(t_n))^2} \qquad (6.2)$$

where K indicates the number of states, N is the total number of observations for each dimension, and $\hat{x}_k(t_n)$ and $x_k(t_n)$ are the predicted and true values for the k-th state at time point t respectively. The RMSEs are very close to each other with the LPMF method having slightly lower error.

Parameter estimation

The estimation results for the parameters of the Lotka-Volterra model are shown in Figure 6.3. The LPMF method again achieves better results than the VGMGP algorithm, even with the positivity constraint on the parameters. The mean values for the prediction from LPMF are also very close to the true parameter values.

To explain this, first note that both methods assume the decoupling of states from the other states and the decoupling of the states and the parameters when constructing the proxy distribution Q. However, the VGMGP method
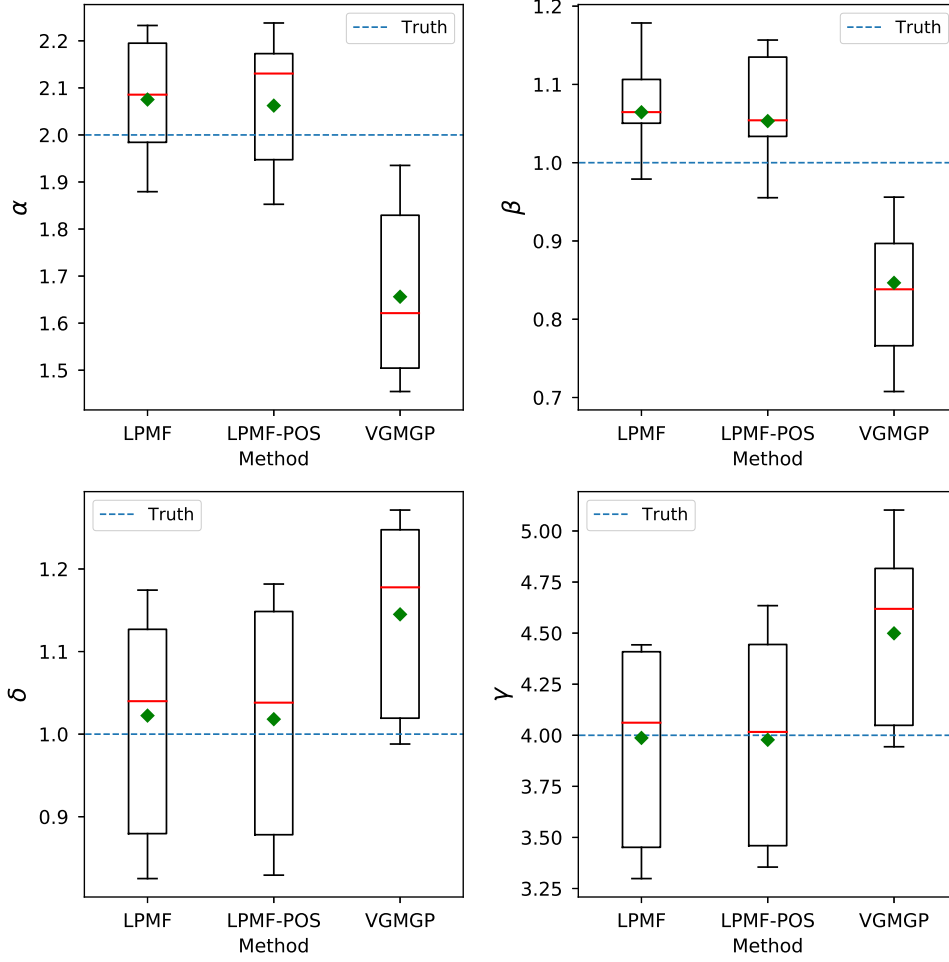
Figure 6.3: Parameter estimation results for the Lotka-Volterra model. In the box plot, the median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles. The true parameter value is shown as the dotted blue line.
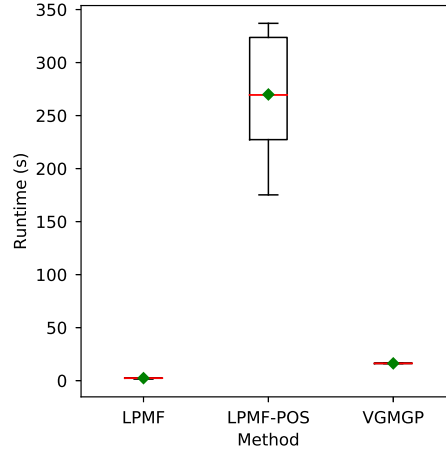
Figure 6.4: Runtime performance for the Lotka-Volterra model. In the box plot, the median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles.

further assumes the decoupling of the same states across time points, which is not the case for LPMF. Since the ODEs of the Lotka-Volterra model satisfies the structural assumption, if the conditional distributions in (Eq. 3.29) and (Eq. 3.30) are indeed Gaussian, then Laplace approximation is expected to correctly find the mode the distribution.

Runtime performance

In terms of runtime, both the LPMF method when no positivity constraint is imposed and the VGMGP method are extremely fast. On average, the LPMF method finishes in 2.4 seconds and the VGMGP method completes in 16.3 seconds. Since LPMF is implemented in Python while VGMGP is implemented MATLAB, exact comparison is infeasible. In general, the LPMF is likely to be more efficient since the evaluation of the expectations (Eq. 3.41) and (Eq. 3.42) is not required. Moreover, the states is inferred by minimizing the cost function for the LPMF method in this experiment. If closed-form solutions are used, it would be expected to even faster.

Lastly, it is unclear what causes the slow down of the LPMF algorithm after the introduction of the positivity constraint. Given the time constraints and in order to achieve fast prototyping, the gradients and Hessians of the cost functions with positivity constraints are obtained from symbolic libraries. This is in contrast to the highly vectorized implementation when no positivity constraint is enforced. Part of the reason is probably due to the inefficient implementation, but it requires further investigation.

Table 6.2: Experimental setup for the protein signaling transduction pathway model. The system dimension is denoted by K and the number of observable dimensions is K_{obs}. Based on the parameters $k_1, k_2, k_3, k_4, V, Km$, the ODEs are solved from time t_0 to t_T with a step size of δt. The observation noise variance σ_k^2 is assumed to be the identical for each state.

K	K_{obs}	t_0	t_T	δt	$k_1, k_2, k_3, k_4, V, Km$	σ_k^2
5	5	0	100	0.05	0.07, 0.6, 0.05, 0.3, 0.017, 3	0.01

6.3 Protein signalling transduction pathway

As already mentioned in Section 1.1, the biochemical *protein signalling transduction pathway* (Vyshemirsky and Girolami, 2007) is a signal transduction cascade model describing the dynamics among protein species. The model can be represented by the following 5-dimensional ODEs:

$$\dot{S} = -k_1 \times S - k_2 \times S \times R + k_3 \times RS$$
$$\dot{dS} = k_1 \times S$$
$$\dot{R} = -k_2 \times S \times R + k_3 \times RS + V \times \frac{Rpp}{K_m + Rpp}$$
$$\dot{RS} = k_2 \times S \times R - k_3 \times RS - k_4 \times RS$$
$$\dot{Rpp} = k_4 \times RS - V \times \frac{Rpp}{K_m + Rpp} \tag{6.3}$$

where the input signal is the concentration level of the protein S, which can either bind to the protein R to form the complex RS, or activate it into its phosphorylated form Rpp, or degrade into dS. The protein Rpp can be deactivated. The conversion between Rpp and R is governed by the *Michaelis-Menten kinetic law* with parameters V and K_m, while the rest of the interactions are defined by the *Mass Action kinetic law* with their respective parameters k_1, k_2, k_3 and k_4.

Different from other dynamical models in this chapter, the state Rpp and the parameter K_m both violate the structural assumption on the ODEs. It is also a difficult benchmark system due to the large number of parameters and its sensitivity to noise. In this section, we use the LPMF method with positivity constraint on the parameters to run the experiment.

Experimental setup

Table 6.2 shows the experimental setup. Since the states flattens out towards the end, the observation time points are set to 0, 1, 2, 4, 5, 7, 10, 15, 20, 30, 40, 50, 60, 80 and 100. Because the observations in this experiment are required to be positive, when the noise is too big to generate negative observations, it is resampled until the positivity requirement is satisfied.
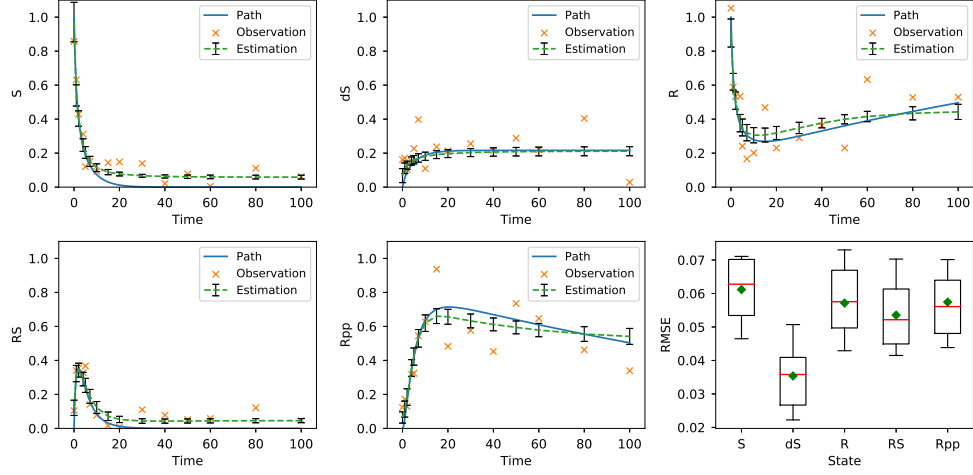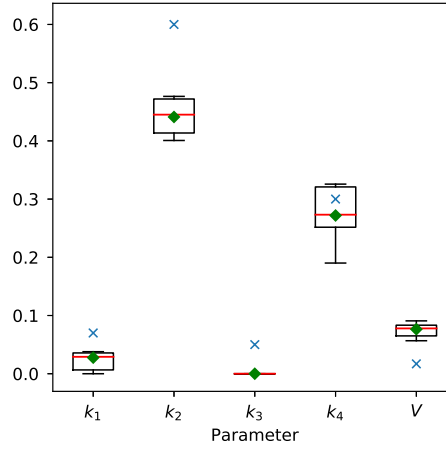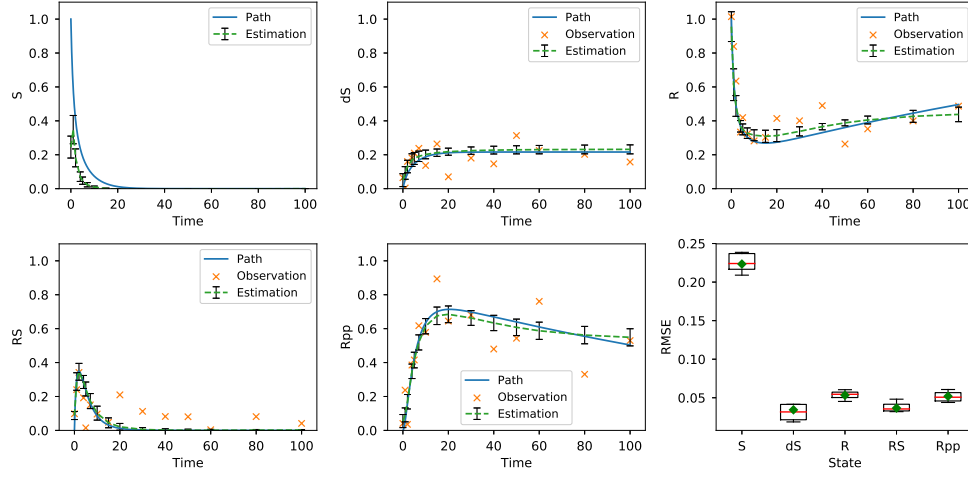
Figure 6.5: State estimation results for the protein signaling transduction pathway model with the parameter K_m set to constant. The ground truth is shown as the blue line. The observations, when available, are shown as the orange crosses. The estimation is shown as the dotted green line. The error bars in the first five plots indicate one standard deviation. For the box plot, the median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles.

Figure 6.6: Parameter estimation results for the protein signaling transduction pathway model with the parameter K_m set to constant. In the box plot, the median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles. The true parameter values are indicated by the blue crosses.
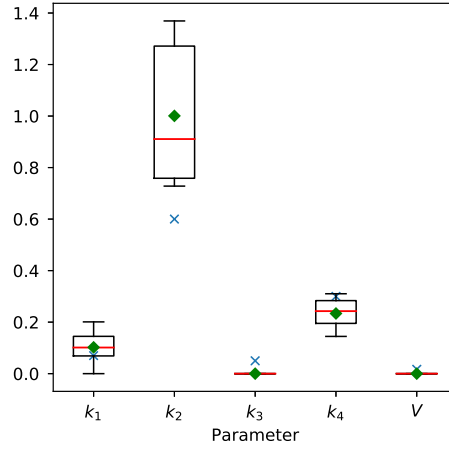
Figure 6.7: State estimation results for the protein signaling transduction pathway model with the parameter K_m set to constant and the state S unobservable. The ground true is shown as the blue line. The observations, when available, are shown as the orange crosses. The estimation is shown as the dotted green line. The error bars in the first five plots indicate one standard deviation. For the box plot, the median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles.

Figure 6.8: Parameter estimation results for the protein signaling transduction pathway model with the parameter K_m set to constant and the state S unobservable. In the box plot, the median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles. The true parameter values are indicated by the blue crosses.

Direct inference on *Rpp*

In this experiment, the parameter K_m is not inferred but all the states are directly inferred. This setup includes the *Rpp* state that violates the structural assumption, which is not possible using the VGMGP method. The results after 10 independent repetitions are shown in Figure 6.5 and Figure 6.6. For state estimation, the overall trend of the states is captured by the learning algorithm. Since there are much fewer observations toward the end of the time point, the prediction becomes worse. The parameter estimation result is not ideal in comparison to the states.

Partial observation

To increase the difficult of the problem, the observations on the state *S* is masked out. The results after 10 independent repetitions is shown in Figure 6.7 and Figure 6.8. Given even less observations, the accuracy of both state and parameter estimation become worse. From Figure 6.7 it can be seen that gradient model indeed does provide some guidance on the estimation of the state *S*.

6.4 Lorenz 96 model

As a minimalistic weather forecast model, the *Lorenz 96* model (Lorenz, 1996) is another widely used benchmark system due to its chaotic behavior under certain configurations and its flexibility to scale to large numbers of states. A *K*-dimensional deterministic Lorenz 96 dynamical system is defined for $k = 1, \dots, K$, state-wise as follows:

$$\dot{x}_k(t) = (x_{k+1}(t) - x_{k-2}(t))x_{k-1}(t) - x_k(t) + F \tag{6.4}$$

where $x_{-1}(t) = x_{K-1}(t)$, $x_0(t) = x_K(t)$, $x_{K+1}(t) = x_1(t)$, and $F \in \mathbb{R}$ is the parameter controlling the behavior of the system. When $F < 0.895$, the states decay into a steady value equal to F; when F is between 0.895 and 4.0, the states are periodic; when $F \geqslant 4.0$, the system exhibits chaotic behavior (Vrettas et al., 2015). An example of the trajectories of a 10-dimensional deterministic Lorenz 96 model is shown in Figure 6.9.

The experiments in this section are mainly concerned about the stochastic Lorenz 96 model, which can be obtained by simply including a noise process as described in Section 2.2. Experiments about its deterministic counterpart can be found in Gorbach et al. (2017). In the following, we compare the results from the LPMF-SDE and the VGPA-MF methods.
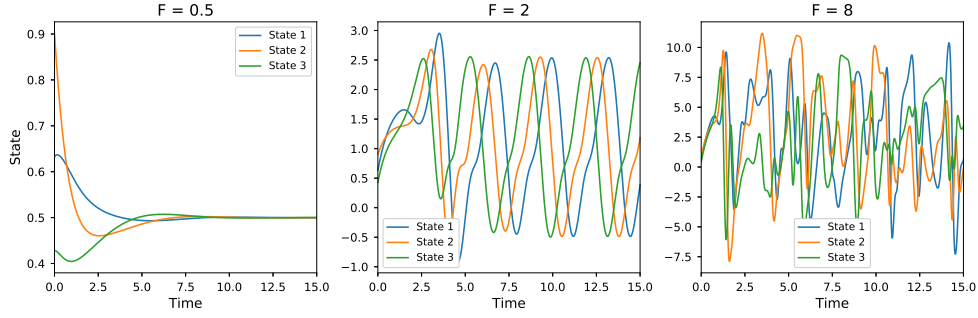
Figure 6.9: Trajectories of the 1st, 2nd and 3rd dimensions of a 10-dimensional deterministic Lorenz 96 model with different parameter values. The numerical integration is carried out from time 0 to 15 with a step size of 0.01.

Table 6.3: Experimental setup to generate sample paths and to collect observations for the stochastic Lorenz 96 model. The system dimension is denoted by K and the number of observable dimensions is K_{obs}. Based on the drift parameter F and the diffusion noise, the sample paths are generated from time t_0 to t_T with a step size of δt. The observation noise variance σ_k^2 and the diffusion noise variance ρ_k^2 are assumed to be the identical for each state. For each time unit, $freq_{obs}$ denotes the number observations to be collected, which are equally distributed over the time line.

K	K_{obs}	t_0	t_T	δt	F	σ_k^2	ρ_k^2	$freq_{obs}$
500	325 (65%)	0	4	0.01	8	1	4	8

Experimental setup

Following Vrettas et al. (2015), Table 6.3 shows the experimental setup to generate sample paths and to collect observations. The sample paths are generated by the VGPA-MF MATLAB code, which uses the first order Euler-Maruyama scheme. To avoid exploiting the special structure of the drift function (Eq. 6.4), the observed states are selected randomly. After observations are collected, the data files are converted into the format compatible with the LPMF-SDE source code.

Note that the system dimension is scaled down from 1000 to 500 due to time constraints. Accordingly, the number of observed states is scaled down proportionally. Nevertheless, it is still very interesting to quantify the scalability of the LPMF-SDE solution, which will be examined in detail in a separate experiment later.

For each SDE sample path, the LPMF-SDE method averages individual inference results from 100 independently generated RODE sample paths to obtain an ensemble solution. The RODE sample paths are estimated in parallel on the ETH Euler cluster[10]. Only 1 CPU core and 2 GB of memory are

---

[10]https://scicomp.ethz.ch/wiki/Euler

allocated to each inference. For this experiment, the RODE states are estimated using the gradient-based method while the parameters are calculated in closed-form.

On the other hand, the VGPA-MF code is deployed on a desktop with an Intel i5 quad-core CPU and 16 GB of memory. We also note that thread pooling is used to utilize all the CPU resources for the VGPA-MF implementation, which is not the case for LPMF-SDE.

State estimation

Figure 6.10 shows an example of the state inference result for the stochastic Lorenz 96 model using both methods. For observed states, the mean values of the predictions from both methods are very close to the ground truth, while the VGPA-MF method tends to have lower variance. For unobserved states, the accuracy of both methods depends on the amount of available information that is directly related to them. As indicated by the drift function (Eq. 6.4), the states are only coupled with their neighbors for the Lorenz 96 model. Therefore, estimation on an unobserved state can be reasonably good when its neighbors are observed, e.g. state 120 has observed neighbors 121, 122, 123, etc. Estimation can fail when the neighbors of that state are also unobserved so that they are unable to provide sufficient information to the algorithm, e.g. the true trajectory of state 213 cannot be recovered since states 211 to 215 are all unobserved. As expected, the uncertainty about the estimation for unobserved states is higher than that for the observed states, which is indicated by the wider error bars.

In order to better quantify the difference between both methods, 10 independent Lorenz 96 SDE sample paths are generated. For each SDE sample path, a set of observations is collected, and then the above experiment is repeated. The root mean square error (RMSE) is used as the accuracy measure, and the RMSEs for observed and unobserved states are considered separately. The formulas for the RMSEs are

$$RMSE_{obs} = \frac{1}{K_{obs}} \sum_{k \in \mathcal{S}_{obs}} \sqrt{\frac{1}{N} \sum_{n=1}^{N} (\hat{x}_k(t_n) - x_k(t_n))^2} \qquad (6.5)$$

$$RMSE_{unobs} = \frac{1}{K_{unobs}} \sum_{k \in \mathcal{S}_{unobs}} \sqrt{\frac{1}{N} \sum_{n=1}^{N} (\hat{x}_k(t_n) - x_k(t_n))^2} \qquad (6.6)$$

where K_{obs} and K_{unobs} indicate the number of observed and unobserved states respectively with $\mathcal{S}_{obs}$ and $\mathcal{S}_{obs}$ as the corresponding sets containing the state indices, N is the total number of observations for each dimension, and $\hat{x}_k(t_n)$ and $x_k(t_n)$ are the predicted and true values for the k-th state at time point t respectively.
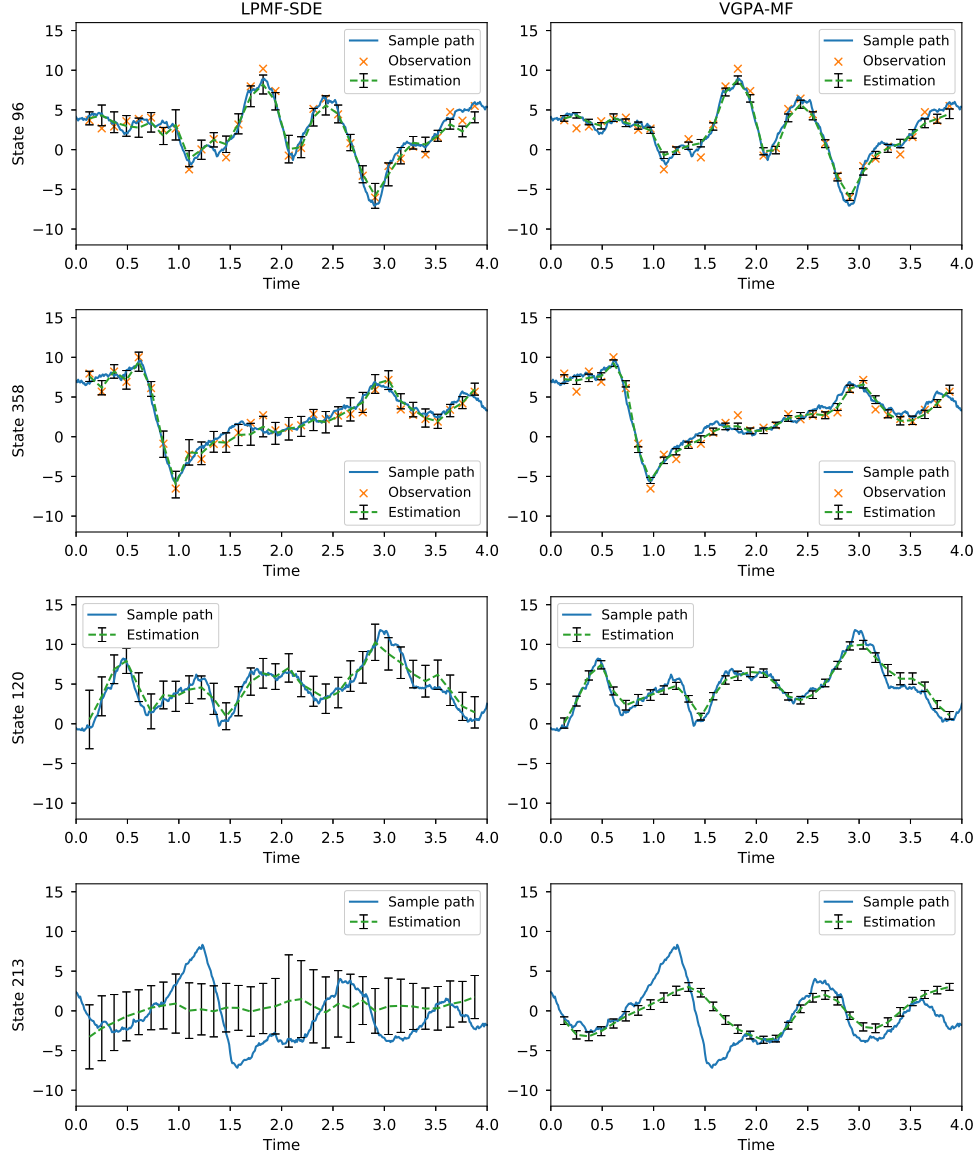
Figure 6.10: State estimation results for 4 selected states from the 500-dimensional stochastic Lorenz 96 model. Among the 500 states, 325 are observed. The left column shows the result using the LPMF-SDE method, while the right column contains the result based on the VGPA-MF algorithm. In the four rows, state 96 and its neighboring states are directly observed; state 358 and only one side of its neighbors are directly observed; state 120 is not observed but its neighbors are; state 213 and its neighbors are all unobserved. The ground truth is shown as a solid blue line. The mean of estimation is drawn as a dotted green line together with the error bars indicating one standard deviation. Observations, when available, are indicated by the orange crosses.
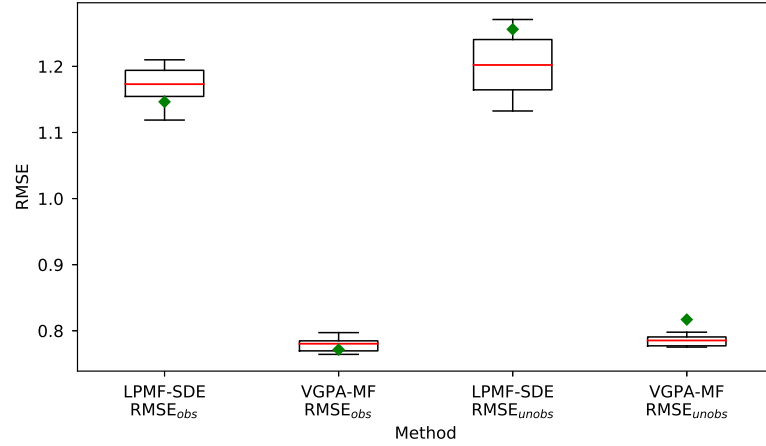
Figure 6.11: Box plot for the RMSEs of state estimation over 10 independent repetitions using sample paths from the 500-dimensional stochastic Lorenz 96 model with 325 observable states. The RMSEs of observed and unobserved states are calculated separately. In the box plot, the median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles.

The results of the 10 independent repetitions are summarized in Figure 6.11. At first glance, it is surprising that the VGPA-MF method has such low RMSE values. This is partially because in the MATLAB code we have obtained, only state inference is carried out and the true drift parameter F is given to the inference procedure. However, the LPMF-SDE method is estimating both the states and parameters simultaneously. Note that the VGPA-MF solution is fully capable of jointly estimating the states and parameters. But as at the time of this writing, we have not obtained the code that computes the gradient of their variational free energy function w.r.t. the parameters in order to complete the required experiments. Moreover, as claimed by Vrettas et al. (2015), their solution can also estimate the diffusion noise, which is an advantage over the LPMF-SDE method and can be considered as a possible future extension to this work. Considering the sparsity of the observations and that the observation noise variances are set to 1, we see some potential for the LPMF-SDE solution.

One shortcoming of the LPMF-SDE method in this experiment is that it only considers a single optimal value of the cost function found using gradient-based method when making predictions. Depending on the initialization conditions and the complexity of the cost function, which is subject to the complexity of drift function, sometimes the optimization procedure can be trapped in a local optimum. As a consequence, even though the mean is recovered reasonably well, the variance of the prediction is less under control. Since currently only out-of-package second-order optimization techniques
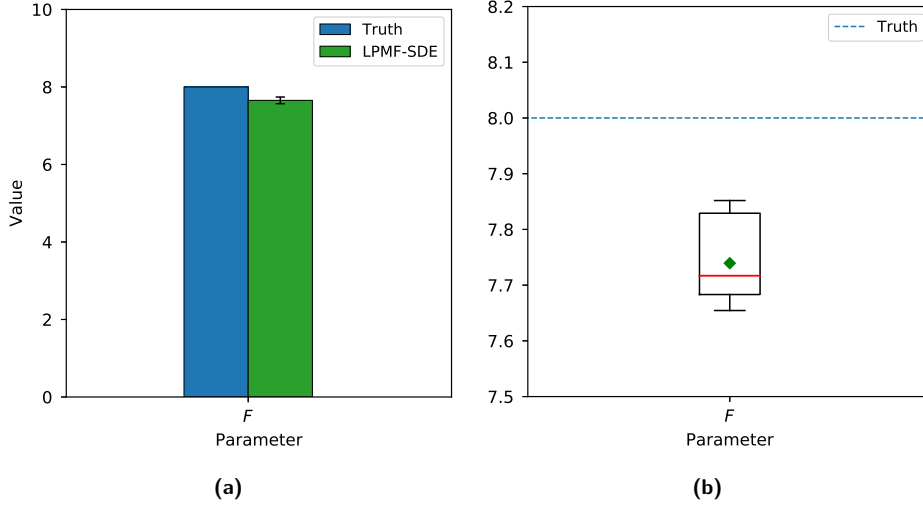
Figure 6.12: Parameter estimation result using the LPMF-SDE algorithm for the 500-dimensional stochastic Lorenz 96 model with 325 observable states. (a) Estimation result for the SDE sample path shown in Figure 6.10. The blue bar indicates the true parameter value. The green bar shows the mean of estimation by averaging individual results from 100 RODE sample paths. The error bar indicates one standard deviation. (b) Box plots for parameter estimation over 10 independent repetitions. In the box plot, the median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles. The dotted blue line indicates the true parameter value.

such as the *truncated Newton method*, the *dog-leg trust-region algorithm*, etc. (Nocedal and Wright, 2006), are used, further improvement on the optimization strategy can also be considered as an extension to this work.

Parameter estimation

This section only discusses parameter inference results. Using the LPMF-SDE algorithm, the result for a single SDE sample path is illustrated in Figure 6.12a, while the result from the 10 dependent repetitions is summarized in Figure 6.12b. Due to the lack of relevant files from the VGPA-MF MATLAB code as mentioned previously, grid search with a step size of 0.5 is performed on one sample path and result is shown in Figure 6.13.

Figure 6.12a shows that the estimation of the drift parameter F is very close to the true value 8, and the variance is much lower in comparison to state estimation. This is due to the fact that the Lorenz 96 model has only one drift parameter, which appears inside the drift function for each state. Hence, the gradient matching model is able to obtain much more information about the drift parameter, despite the existence of unobserved states. Figure 6.12b further shows that the mean values of parameter estimation over the 10 inde-
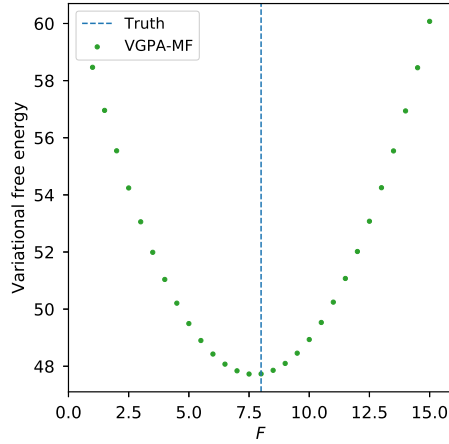
Figure 6.13: Grid search result using the VGPA-MF algorithm for the stochastic Lorenz 96 model using sample paths shown in Figure 6.10. The green dots indicate where the grid search is carried out, while the dash blue line indicates the true parameter value.

pendent repetitions are narrowly distributed. But in general, the LPMF-SDE method tends to underestimate the parameter value. Lastly, although Figure 6.13 shows that the variational free energy of the VGPA-MF algorithm reaches the minimum at the true parameter value, but the whole experiment took almost 36 hours to complete. For dynamical system with several parameters, the combinatorial nature of the parameters would render this approach impractical.

Runtime performance

For the SDE sample path shown in Figure 6.10, the average runtime for the LPMF-SDE algorithm to solve one RODE sample path is 2309 seconds with a standard deviation of around 128 seconds, while the runtime for the VGPA-MF method takes 3232 seconds. As the current VGPA-MF MATLAB code only runs the state smoothing algorithm, it is expected to be much more computationally intensive when parameter estimation is carried out simultaneously, since the forward-backward loop has to be entered many times until convergence.

Figure 6.14 summarizes the average runtimes for LPMF to solve one RODE sample path and the runtimes for VGPA-MF to complete one iteration of state smoothing over the 10 independent repetitions. It shows that the LPMF-SDE solution performs consistently faster and more stably than its counterpart, as indicated by lower median runtime value and the narrower distribution.

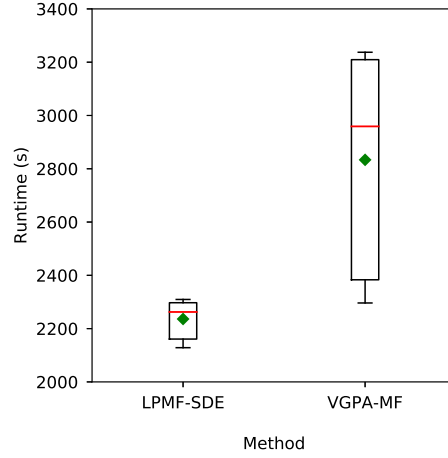To conclude, the LPMF-SDE algorithm requires much less resources and

Figure 6.14: Box plot for the runtimes over 10 repetitions using independent sample paths from the stochastic Lorenz 96 model. The median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles.

also runs faster for inference on one RODE sample path. However, the ensemble nature of the LPMF-SDE strategy relies on averaging results from a large number RODE estimations, which could increase the computational requirements significantly. Nevertheless, given the availability of computer farms and the ease of setting up distributed inference pipeline nowadays, the parallelism requirement for the LPMF-SDE solution can be easily satisfied.

Scalability

As mentioned by Gorbach et al. (2017), inference on the deterministic Lorenz 96 model with 1000 states using their VGMGP solution completes within 400 seconds on average. However, the LPMF-SDE method, which extends their methodology, takes on average more than 2200 seconds for one RODE sample path with only 500 states, as shown in Figure 6.14. It is therefore interesting to re-examine the scalability of the LPMF-SDE solution.

First, we noticed during implementation that the performance of the LPMF method for ODEs is comparable to the VGMGP method. The major difference between LPMF-SDE and LPMF is the introduction of a stochastic Ornstein-Uhlenbeck process into the vector field of the ODEs. To reveal the influence of the stochastic process on the inference procedure, the following experiment compares side-by-side the performance of the LPMF-SDE method running on the stochastic Lorenz 96 model with the performance of the LPMF method running on its deterministic counterpart as the number of states increases.
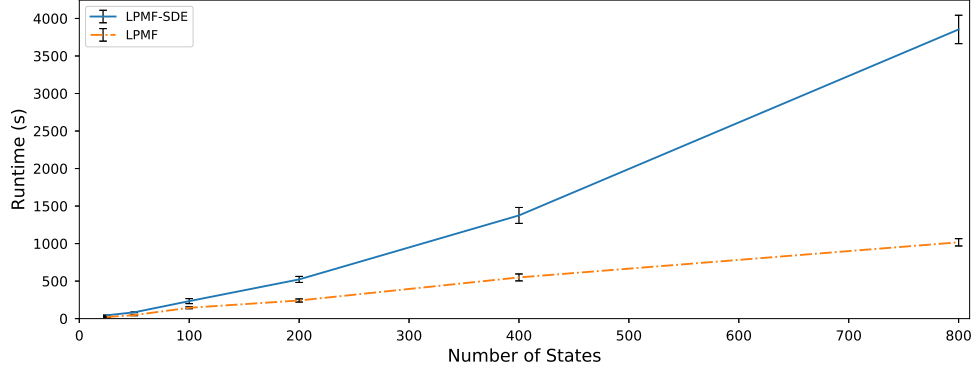
Figure 6.15: Scalability of the LPMF-SDE algorithm versus its deterministic counterpart, i.e. LPMF, as the dimensionality of the system increases. For the LPMF-SDE method, the blue line indicates the average interpolated runtime to inference one RODE sample path over 10 independent RODE sample paths by connecting the measurements at 25, 50, 100, 200, 400 and 800. For the LPMF method, the orange line is obtained by connecting the averages of 10 independent runs on the same ODE trajectory with different observations at the same measurement points. The error bars in both setups indicate one standard deviation.

Specifically, measurements are taken with a system dimensionality of 25, 50, 100, 200, 400, and 800 with the number of observed states kept at 65 percent of the total number of states. Using the same setup in Table 6.3, except that the diffusion noise is not used for the Lorenz 96 ODEs, the LPMF-SDE algorithm is tested with 10 independent RODE sample paths while the LPMF method is tested with 10 independent observation sets from the same ODE trajectory for each system dimensionality. The experiments are all conducted on the Euler cluster with the same hardware configuration.

As can be seen from Figure 6.15, benefiting from the efficient implementation of the gradient and Hessian evaluation subroutines, the gradient-based LPMF method incurs only a slightly higher performance penalty than the closed-form solution from VGMGP. Also, the runtime seems to increase linearly as the number of states increases. On the other hand, the performance of the LPMF-SDE method degrades much faster than LPMF, and the gap between them become larger as the number of states increases. The cause of this phenomenon requires further investigation but a plausible explanation is the complication of the optimization objective after the introduction of the stochastic process into the vector field.

6.5 Lorenz 63 model

The last dynamical system examined in this chapter is the stochastic version of the *Lorenz 63* model (Lorenz, 1963), which is a low dimensional mathematical model for thermal convection in the atmosphere. The vector field of the deterministic Lorenz 63 is defined as follows:

$$f(x(t), \theta) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} \sigma(y(t) - x(t)) \\ x(t)(\rho - z(t)) - y(t) \\ x(t)y(t) - \beta z(t) \end{bmatrix} \tag{6.7}$$

where the state vector is given by $x(t) = [x(t), y(t), z(t)]^\top \in \mathbb{R}^3$, and $\theta = [\sigma, \rho, \beta]^\top \in \mathbb{R}^3$ is the parameter vector controlling the system behavior. Although consisting of only 3 states, this model is highly nonlinear and exhibits chaotic behavior under certain parameter configurations.

Correspondingly, the stochastic Lorenz 63 model with a state-specific, additive noise process is given by

$$dx(t) = f(x(t), \theta)dt + \Sigma^{\frac{1}{2}}dW_t \tag{6.8}$$

where $f(x(t), \theta)$ is defined in (Eq. 6.7), $\Sigma \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix containing the diffusion noise variance, and W_t is a 3-dimensional standard Wiener process. Using the parameter set $[10, 28, \frac{8}{3}]^\top$, which is well-known for its resulting chaotic behavior, a sample path is shown in Figure 6.16.

Comparison algorithm

In order to make comparison, a minimalistic MAP estimation of the drift parameters based on (Vrettas et al., 2011, Table 3) is self-implemented[11] by extending the Python source code[12] for the VGPA algorithm. In the following, the abbreviation VGPA-MAP is used to refer to this extension.

The extension consists of an inner loop and an outer loop. The inner loop enhances state estimation by running the VGPA smoothing algorithm to compute the optimal approximate state posterior, based on the current estimation of the parameters. The outer loop then takes a gradient step to update the parameters. This procedure is repeated until either the gradient over the parameters vanishes or the state estimation from the inner loop does not improve further significantly. Details of the algorithm can be found in (Vrettas et al., 2011, Section 5.2).

Note that Vrettas et al. (2011) claim that the aforementioned algorithm can be similarly applied to the estimation of the diffusion noise covariance matrix

---

[11]https://github.com/ruiixu23/VGPA
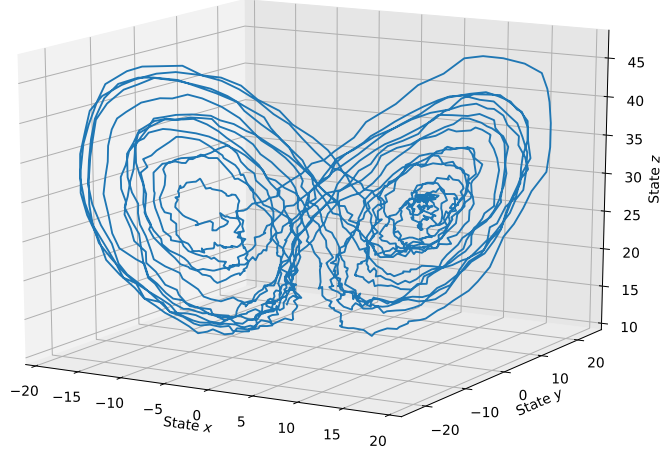[12]https://github.com/vrettasm/VGPA

Figure 6.16: Sample path for the stochastic Lorenz 63 model generated based on the parameter values $[\sigma, \rho, \beta]^\top = [10, 28, \frac{8}{3}]^\top$ and the diffusion noise variances $\rho_1^2 = \rho_2^2 = \rho_3^2 = 10$. The integration is performed from time 0 to 20 with a step size of 0.01.

Σ, which is not implemented here as the current LPMF-SDE method only supports inference on the drift parameters. Also, the extension adopts a simple gradient update strategy with a fixed learning rate, which may lead to suboptimal result but can nonetheless be used as a baseline.

Experimental setup

Adopting the experimental setup from Vrettas et al. (2015), the configuration used to generate sample paths and to collect observations is shown in Table 6.4. Specifically, the sample paths are generated by the VGPA source code using the first order Euler-Maruyama method with a small step size to achieve higher accuracy. After observations are collected, the data files are transformed into the format compatible with the LPMF-SDE method.

In order to provide a fair comparison, both methods are deployed to the Euler cluster using the same hardware configuration described in Section 6.4. For LPMF-SDE, 100 independent RODE sample paths are solved to obtain an ensemble result for one SDE sample path. The states are optimized using a gradient-based method while the parameters are calculated analytically with mirroring of negative parameters[13]. For VGPA-MAP, the learning rate is fixed to 0.001, the maximum number of iterations is 250, and the stopping

---

[13]The mirroring of negative parameters is a heuristic solution. A better approach would be the enforcement of positivity constraint.

Table 6.4: Experimental setup to generate sample paths and to collect observations for the stochastic Lorenz 63 model. The system dimension is denoted by K and the number of observable states is denoted by K_{obs}. Given the drift parameters σ, ρ, β and the diffusion noise, the sample paths are generated from time t_0 to t_T with a step size of δt. The variances of the observation noise σ_k^2 and the diffusion noise ρ_k^2 are assumed to be identical for each state. For each time unit, $freq_{obs}$ denotes the number of observations to be collected, which are equally distributed over the time line.

K	K_{obs}	t_0	t_T	δt	σ, ρ, β	σ_k^2	ρ_k^2	$freq_{obs}$
3	3 or 2	0	20	0.01	$10, 28, \frac{8}{3}$	2	10	5

criteria are the decrease in the total variational free energy and the *L2 norm* of the change to the drift parameters with a threshold of 10^{-6}. We noticed that all the experiments from VGPA-MAP reached the maximum number of 250 iterations. After manual examination, the result at the 80th iteration is used for the following comparisons.

It would be interesting to compare these two methods with one unobserved state. Unfortunately, the smoothing result from the VGPA algorithm with unobserved state is not ideal from preliminary examination. Therefore, a comparison with full state observability is provided. But in order to demonstrate that the LPMF-SDE solution is capable of handling partial observations, an additional experiment is conducted by masking out the observations for state y. To distinguish the two scenarios, we use LPMF-SDE-F and LPMF-SDE-P to refer to the fully and partially observable cases respectively.

State estimation

The estimation for state y from one SDE sample path is shown in Figure 6.17. Similar to the result for the stochastic Lorenz 96 model in Section 6.4, the general trend of the state is sucessfully captured by both methods with minor differences on the details. The VGPA-MAP method seem to be better at points where the state changes dramatically to form high and low peaks. The variance of the estimation is also lower than the LPMF-SDE approach. One the other hand, it is worth noting that the LPMF-SDE method even functions quite well when state y is unobserved but with naturally higher uncertainty, especially around the peaks. Since state y is unobserved, the only source of information to estimate it is from the drift function (Eq. 6.7), which is demonstrates the potential of the gradient matching framework.

After repeating the above experiment 10 times using each time an independently generated SDE sample path and observation set, the RMSEs for state estimation is computed and summarized in Figure 6.18. Since there are only three states, the RMSEs over all states are considered together. The figure shows that the RMSE of state estimation using VGPA-MAP is slightly below the observation noise variance, which outperforms LPMF-SDE with full
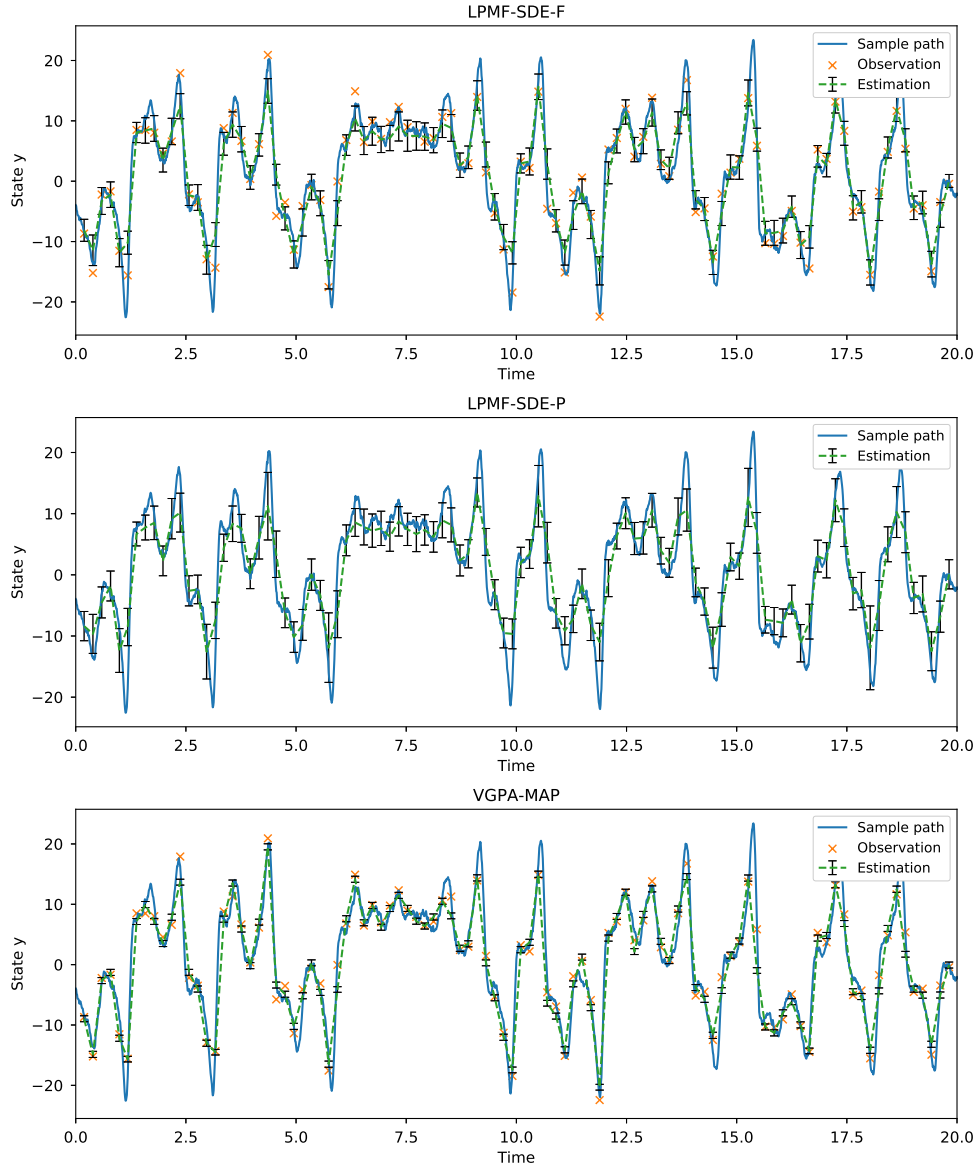
Figure 6.17: Estimation for state y of the stochastic Lorenz 63 model. The top and middle plots are the results for the fully and partially observable cases using the LPMF-SDE method respectively. The bottom plot contains the result for the fully observable case using the VGPA-MAP extension. The ground truth is shown as a solid blue line. The mean of the estimation is drawn as a dotted green line together with the error bars indicating one standard deviation. Observations, when available, are indicated by the orange crosses.
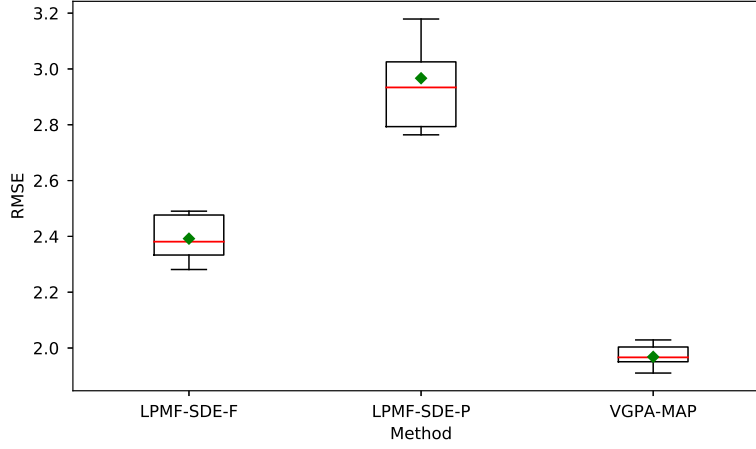
Figure 6.18: Box plot for the RMSEs of state estimation over 10 independent SDE sample paths from the stochastic Lorenz 63 model. In the box plot, the median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles.

state observability by around 0.4 on average. On the other hand, the RMSE using LPMF-SDE when state y is unobserved is still reasonably low, considering that the values of the state range from -20 to 20, and the inference is carried out over a much longer time period than all previous experiments.

Parameter estimation

For parameter estimation, the results are presented in Figure 6.19. Overall, the estimation based on both methods seem to be on par with each other with relatively low variance when all the states are observed. Given that the VGPA-MAP algorithm achieves better accuracy when estimating the states, it would be expected that the parameter estimation to also be better. The non-optimal performance is probably due to the simple gradient update strategy as mentioned before.

If we look at Figure 6.19a in detail, the large variance around the estimation for parameter σ when state y is unobserved is noticeable. To explain this, first note that σ appears only inside the first equation of (Eq. 6.7) together with state x and y. Since y is not observable, the only source of information to estimate σ is x. This is in contrast to the other two parameters ρ and β, where both states x and z are used to estimate them. Lastly, the means of the predicted parameters over the 10 independent runs are generally concentrated except for parameter ρ inferred by the VGPA-MAP algorithm, as shown in Figure 6.19b to Figure 6.19c.

To further illustrate the difficulty of parameter estimation, grid search on
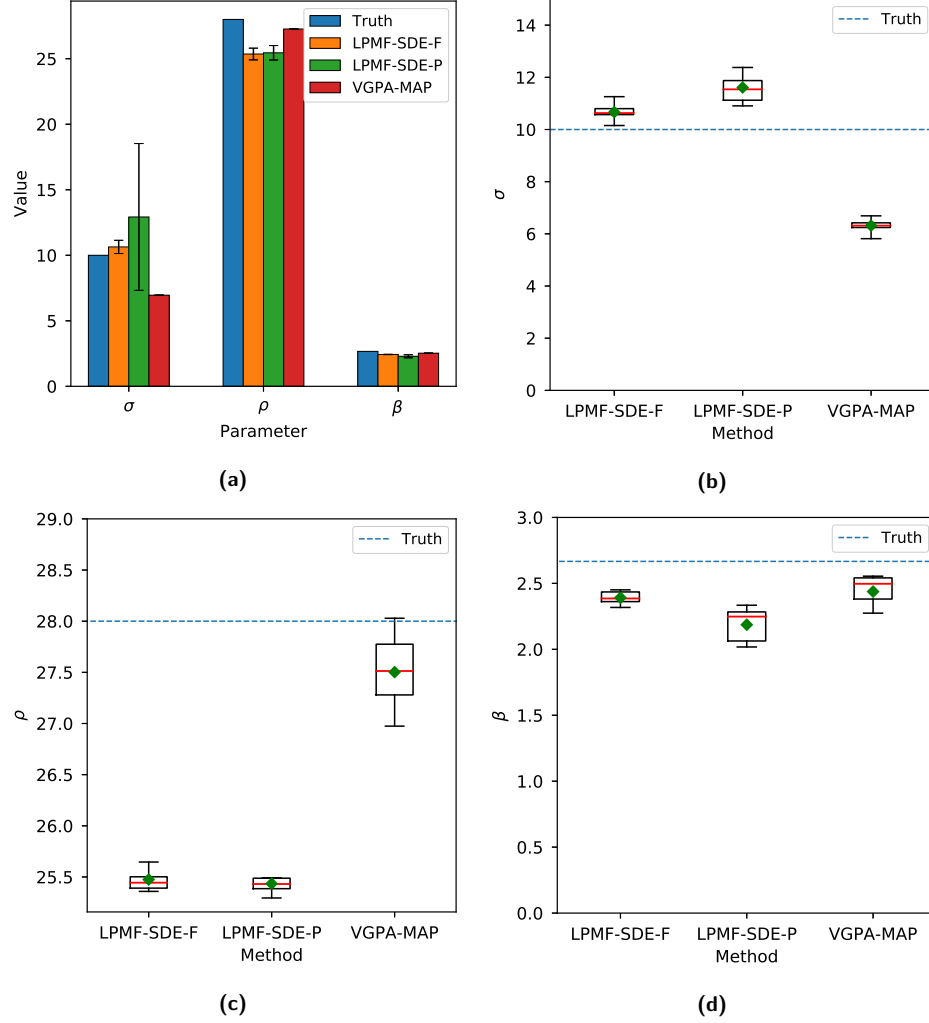
Figure 6.19: Parameter estimation result for the stochastic Lorenz 63 model. (a) Estimation result for the SDE sample path shown in Figure 6.17 with the error bar indicating one standard deviation. (b), (c), and (d) are the box plots for parameters σ, ρ and β after 10 independent repetitions respectively. In the box plot, the median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles. The dotted blue line indicates the true parameter value.
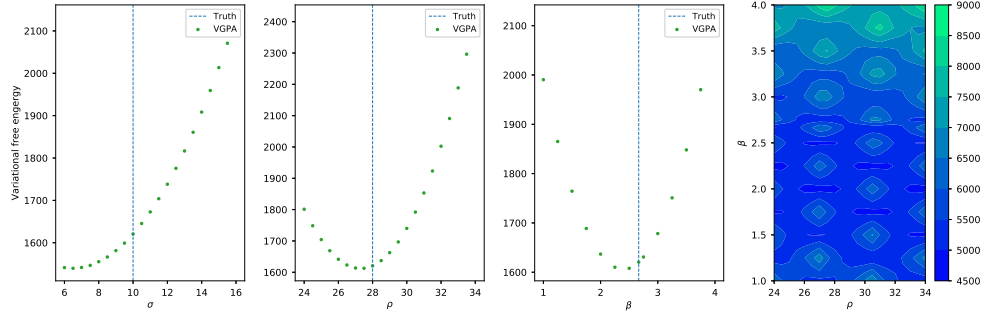
Figure 6.20: Grid search result using the VGPA algorithm for the stochastic Lorenz 63 model. The green dots indicate where the grid search is carried out, while the dash blue line indicates the true parameter value. The color map shows the existence of local minima when varying the parameters β and ρ together.
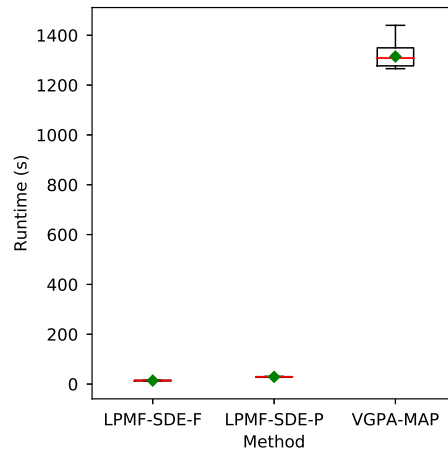

Figure 6.21: Box plot for the runtimes over 10 repetitions using independent sample paths from the stochastic Lorenz 63 model. The median is indicated by the red line, while the mean is shown as the green diamond. The box shows the lower and upper quartiles, while the whiskers are the 5th and 95th percentiles.

the parameters is conducted using the original VGPA method. Specifically, the step size used for the grid search is 0.5. The ranges for the searches for σ, ρ, β are $[6, 16]$, $[24, 34]$ and $[1, 4]$ respectively. Using the first plot in Figure 6.20 as an example, the changes in the variational free energy is plotted as the parameter σ changes while the other two parameters are kept at the true values. It can be seen that the parameter value that leads to the minimum of the variational free energy deviates from the true parameter values. Moreover, the last plot in Figure 6.20 shows the existence of many local minima as we vary two parameters together, which makes parameter estimation based on grid search difficult.

Runtime performance

Figure 6.21 shows the distribution of the average time to solve one RODE sample path and the runtime of the VGPA-MAP algorithm to infer one SDE sample path after 10 independent repetitions. For each RODE sample, LPMF-SDE takes a fraction of the time required by VGPA-MAP for one SDE sample path. Even if the RODEs are not solved in parallel, the total runtime required by LPMF-SDE is still comparable to that of VGPA-MAP. This further demonstrates the runtime efficiency of the LPMF-SDE scheme.

Chapter 7

Conclusion

This work examines the problem of state and parameter estimation in deterministic and random dynamical systems given noisy, sparse or even incomplete observations. A mean-field Laplace approximation solution is proposed to address the intractability of the posterior distribution and has been shown empirically to be robust and scalable. Further, it relaxes the structural assumption imposed on the dynamical systems from previous work and introduces positivity constraints on the states and parameters. Based on the correspondence between RODEs and SDEs, a highly efficient parallel inference technique is devised to address problems involving diffusion processes.

Bibliography

Archambeau, C., Cornford, D., Opper, M., and Shawe-Taylor, J. Gaussian process approximations of stochastic differential equations. In *Gaussian Processes in Practice*, pages 1–16, 2007.

Archambeau, C., Opper, M., Shen, Y., Cornford, D., and Shawe-taylor, J. S. Variational inference for diffusion processes. In *Advances in Neural Information Processing Systems*, pages 17–24, 2008.

Babtie, A. C., Kirk, P., and Stumpf, M. P. H. Topological sensitivity analysis for systems biology. *Proceedings of the National Academy of Sciences*, 111(52): 18507–18512, 2014.

Bishop, C. M. *Pattern recognition and machine learning*. Springer, 2006.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112 (518):859–877, 2017. doi: 10.1080/01621459.2017.1285773.

Butcher, J. C. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

Calderhead, B., Girolami, M., and Lawrence, N. D. Accelerating bayesian inference over nonlinear differential equations with gaussian processes. In *Advances in neural information processing systems*, pages 217–224, 2009.

Dondelinger, F., Husmeier, D., Rogers, S., and Filippone, M. Ode parameter inference using adaptive gradient matching with gaussian processes. In *Artificial Intelligence and Statistics*, pages 216–228, 2013.

Ellner, S. P. and Guckenheimer, J. *Dynamic models in biology*. Princeton University Press, 2011.

Gardiner, C. W. *Stochastic methods: a handbook for the natural and social sciences*. Springer, 2009.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Gorbach, N. S., Bauer, S., and Buhmann, J. M. Mean-field variational inference for gradient matching with gaussian processes. *arXiv preprint arXiv:1610.06949*, 2016.

Gorbach, N. S., Bauer, S., and Buhmann, J. M. Scalable variational inference for dynamical systems. *arXiv preprint arXiv:1705.07079*, 2017.

Grüne, L. and Kloeden, P. Pathwise approximation of random ordinary differential equations. *BIT Numerical Mathematics*, 41(4):711–721, 2001.

Higham, D. J. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546, 2001.

Hinton, G. E. Products of experts. In *ICANN'99*, volume 1, pages 1–6, 1999.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Imkeller, P. and Schmalfuss, B. The conjugacy of stochastic and random differential equations and the existence of global attractors. *Journal of Dynamics and Differential Equations*, 13(2):215–249, 2001.

Jensen, J. L. W. V. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30(1):175–193, 1906.

Jentzen, A. and Kloeden, P. E. *Taylor approximations for stochastic partial differential equations*. SIAM, 2011.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Kalnay, E. *Atmospheric modeling, data assimilation and predictability*. Cambridge university press, 2003.

Kloeden, P. E. and Jentzen, A. Pathwise convergent higher order numerical schemes for random ordinary differential equations. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 463, pages 2929–2944. The Royal Society, 2007.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(14):1–45, 2017.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Lorenz, E. The butterfly effect. *World Scientific Series on Nonlinear Science Series A*, 39:91–94, 2000.

Lorenz, E. N. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

Lorenz, E. N. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, 1996.

Lotka, A. J. The growth of mixed populations: two species competing for a common food supply. *Journal of the Washington Academy of Sciences*, 22 (16/17):461–469, 1932.

Macdonald, B. and Husmeier, D. Gradient matching methods for computational inference in mechanistic models for systems biology: a review and comparative analysis. *Frontiers in bioengineering and biotechnology*, 3, 2015.

MacKay, D. J. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Nocedal, J. and Wright, S. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006. ISBN 9780387227429.

Øksendal, B. *Stochastic Differential Equations: An Introduction with Applications*. Springer, 2013.

Petersen, K. B. and Pedersen, M. S. The matrix cookbook. 2012.

Ramsay, J. O., Hooker, G., Campbell, D., and Cao, J. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796, 2007.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian processes for machine learning*, volume 1. MIT press, 2006.

Riesinger, C., Neckel, T., and Rupp, F. Solving random ordinary differential equations on gpu clusters using multiple levels of parallelism. *SIAM Journal on Scientific Computing*, 38(4):C372–C402, 2016.

Sussmann, H. J. On the gap between deterministic and stochastic ordinary differential equations. *The Annals of Probability*, pages 19–41, 1978.

Vrettas, M. D., Cornford, D., and Opper, M. Estimating parameters in stochastic systems: A variational bayesian approach. *Physica D: Nonlinear Phenomena*, 240(23):1877–1900, 2011.

Vrettas, M. D., Opper, M., and Cornford, D. Variational mean-field algorithm for efficient inference in large systems of stochastic differential equations. *Physical Review E*, 91(1):012148, 2015.

Vyshemirsky, V. and Girolami, M. A. Bayesian ranking of biochemical system models. *Bioinformatics*, 24(6):833–839, 2007.

Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

_____

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

SCALABLE VARIATIONAL INFERENCE FOR STOCHASTIC DIFFERENTIAL EQUATIONS

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
XU	RUIFENG

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 01.09.2017	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.