Inference for Continuous Stochastic Processes Using Gaussian Process Regression

by

Yizhou Fang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Statistics

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Gaussian process regression (GPR) is a long-standing technique for statistical interpolation between observed data points. Having originally been applied to spatial analysis in the 1950s, GPR offers highly nonlinear predictions with uncertainty adjusting to the degree of extrapolation – at the expense of very few model parameters to be fit. Thus GPR has gained considerable popularity in statistical applications such as machine learning and non-parametric density estimation. In this thesis, we explore the potential for GPR to improve the efficiency of parametric inference for continuous-time stochastic processes. For almost all such processes, the likelihood function based on discrete observations cannot be written in closed-form. However, it can be very well approximated if the inter-observation time is small. Therefore, a popular strategy for parametric inference is to introduce missing data between actual observations. In a Bayesian context, samples from the posterior distribution of the parameters and missing data are then typically obtained using Markov chain Monte Carlo (MCMC) methods, which can be computationally very expensive. Here, we consider the possibility of using GPR to impute the marginal distribution of the missing data directly. These imputations could then be leveraged to produce independent draws from the joint posterior by Importance Sampling, for a significant gain in computational efficiency. In order to illustrate the methodology, three continuous processes are examined. The first one is based on a neural excitation model with a non-standard periodic component. The second and third are popular financial models often used for option pricing. While preliminary inferential results are quite promising, we point out several improvements to the methodology which remain to be explored.

Acknowledgements

The path of writing this thesis is full of challenges but also full of surprises. It is the very first time that I learn the frustration and joy along the way of research. Anyway, it is a great experience.

Most of all, I cannot overcome those ups and downs without my dear supervisor Professor Martin Lysy, who is full of brilliant ideas and always be there when I most needed him. I cherish all his patience and encouragements, which are my great propulsion.

I own thanks to all the people in the master office. Yoshi and Lin, who are also working on their theses, I'm thankful for all the inspiring talks that support me through.

Finally, I thank my parents for their unconditional support, and my grandmother, who always be there for me. I am also grateful to my uncle, without whose guidance starting since I was a child, I cannot achieve what I am right now.

Dedication

This is dedicated to my parents, who I love deeply, and to my future doctoral years.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 A Brief History

Gaussian process regression (GPR) is a statistical technique for interpolating between data observations. Unlike least-squares regression methods, GPR does not specify a conditional mean function but rather the covariance function between responses. In addition, GPR can easily achieve nonlinear predictions and prediction uncertainty which naturally accounts for the degree of extrapolation from the observed data – both of which are difficult for mean-based regression models such as least squares.

GPR is not a recent topic in statistics. Its origins can be traced back to the works of Wiener and Kolmogorov in the 1940's [41]. GPR is also known as"Kriging" after Danie G. Krige [18], the first person to apply GPR in a statistical context: to estimate the gold distribution at unknown locations based on observated borehole samples. Based on Krige's work, George Matheron [24] developed the theory of GPR and promoted its widespread use in the field of geostatistics. A comprehensive review of GPR in geostatistics was written by Cressie [7] with more recent works by Diggle [9] and Delfner [8]. The R package 'geor' [35] is also devoted to GPR and 'large data' spatial problems are discussed in [1].

Unlike least-squares regression, GPR easily and naturally produces highly non-linear predictions with heteroscedastic confidence intervals. As such, it has received considerable attention from other statistical fields. For example, GPR is commonly used in the machine learning community. Many of the early developments had a Bayesian perspective, starting with the works of Neal [26], Rasmussen [34] and Williams [39]. The book "Gaussian Process for Machine Learning" written by Rasmussen and Williams [31] summarized the

previous work and considerably elaborated on the subject of covariance function modelling. Gaussian process also serves as a method for classification in machine learning community. Williams and Barber [40] provided an introduction to Gaussian Process Classification and its comparison with other methods was discussed by Kuss and Rasmussen [19].

Another important application of GPR is to meta-modelling of computer experiments. Computer experiments usually contain various inputs and every run with the code might be computationally expensive. Finding a predictor based on stochastic processes can be a relatively cheap and efficient choice. An application to aircraft simulations is given in [36], and other examples are given in [23] and [14].

GPR is also recognized as an important alternative choice for neural network models. During the 1990's, various comparisons between neural networks and GPR were explored by Rasmussen [34] and Neal [27], the conclusion being that GPR with limited parameters could avoid problems such as overfitting. Biological applications of GPR to learning and prediction for large batch datasets, including functional and longitudinal data, have been proposed by [38] and [28]. In fact, GPR is becoming increasingly popular for modeling very general nonlinear dynamic systems [25], with positive results for robustness discussed in [16].

In this thesis, we shall explore the potential of GPR to improve the efficiency of parametric inference for continuous-time stochastic processes. Such processes are almost always observed in discrete time, in which case the likelihood function cannot be written in closed form. However, it can be increasingly well approximated as the time between observations decreases to zero. Thus, a popular inference strategy is to add missing data between actual observations. In a Bayesian context, samples from the joint posterior distribution of missing data and parameters are obtained by Markov chain Monte Carlo (MCMC) techniques, which can be computationally very expensive [17]. The bottleneck occurs from the large amount of missing data required for practical applications, as the low-dimensional parameter draws conditioned on the complete data are relatively simple to produce. Thus, we propose to first marginally impute the missing data based on the actual observations using GPR techniques. In combination with the above mentioned parameter draws, this produces independent proposals from the joint posterior distribution which can be used for importance sampling. If these proposals are close to the targeted posterior, then our Monte Carlo inference will be vastly more efficient than MCMC.

1.2 Overview

Chapter 2 will introduce the methodology of GPR along with a simple example illustration. Basic definitions will be included and the choice of GPR covariance function and parameter estimation will be discussed. We will also talk about some computational advantages about estimating the parameters. Multi-responses GPR will be briefly introduced too.

In Chapter 3, we use GPR to impute missing observations under the well-known FitzHugh-Nagumo model (FHN). We find the non-standard periodicity of the FHN to be remarkably well-captured by the GPR with a single sinusoidal component in the covariance.

In Chapter 4 we turn to inference for continuous-time stochastic processes using two well-known financial models. The first is the Cox-Ingersoll-Ross (CIR) diffusion process, for which GPR produces excellent estimates of the missing data distribution. The second process is Heston's stochastic volatility model, a highly non-Gaussian bivariate process on which the importance sampling methodology is tested out. While preliminary inferential results are quite promising, in Chapter 5 we point out several improvements to the methodology which remain to be explored.

Chapter 2

Gaussian Process Regression

2.1 Motivation

Consider a simple regression problem based on the data points shown in Figure 2.1. Seven observation points $\mathbf{y}$ based on independent variable $\mathbf{t}$ and two points we want to estimate when we are given the $\mathbf{t}$ values of these points [1]. Point $t = 2.3$ shows a typical situation when we want to estimate a missing point between observations. Point $t = 5.5$ represents a typical situation when a future prediction is wanted. So what is our strategy to estimate $\mathbf{y}$ value at these locations?

Arguably, a default strategy would be to apply ordinary least square (OLS) regression models and we can see the result from 2.1. However, a linear model seems like a bad choice because there lacks a linear trend in the observations. The quadratic model can be really tempting, but some observation points are way off track. The cubic model can be a good choice here. However a cubic model needs a sharp trend going up before $t = 1$ and a sharp trend going down after $t = 5$, which we cannot determine from the observation data. If there are no these properties, cubic model can be dangerous to use as cubic term changes so quick that the prediction can be far from the true value.

Gaussian Process Regression (GPR) provides a different approach to manage this estimation problem. Unlike the conventional models make assumptions about the true models with error term, GPR assumes observations are coming from a stochastic Gaussian process. Technically, GPR estimation is not through a prediction function $f(t)$ rather a conditional distribution at the point t given the observation points. The conditional mean can serve

[1] This variable $\mathbf{t}$ can always be the time line, e.g. daily data, monthly data.

as a good point estimation and confidence interval follows naturally from the conditional mean and variance.

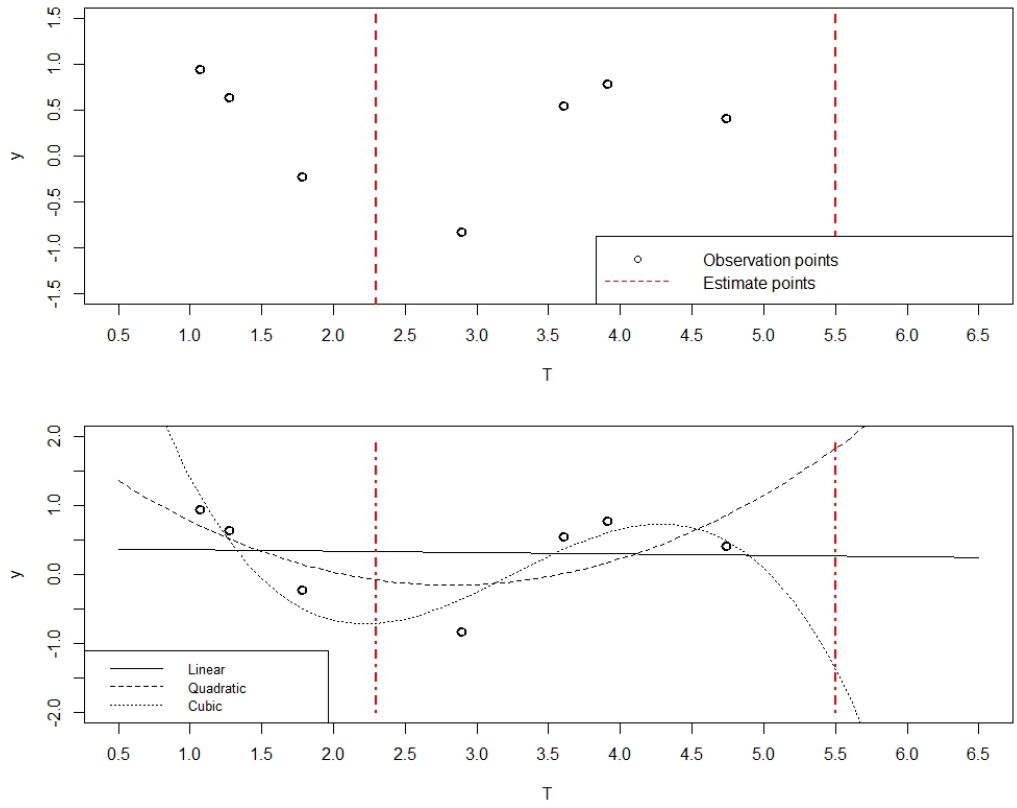

Figure 2.1: A Simple Demonstration

GPR has several advantages over conventional regression model:

1. GPR seems a complex method, but it does not require too many parameters. Simplest setting only needs two parameters, which is equal to the simplest linear regression model. But GPR can generate non-linear estimation based on so less parameters.

2. GPR has prediction uncertainty which naturally accounts for the degree of extrapolation from the observed data. The nearer the prediction points are to the observations, the narrower the confidence intervals. Although OLS prediction also has heteroscedastic property, but it does not has the property mentioned above as GPR.

3. GPR has a highly non-linear estimation curve. Speaking from the prediction aspect, this advantage is quite important, because this means the estimation is more flexible and more fit with the observation points.

4. Noisy observations can fit right into GPR. We will find out in the following part that GPR's covariance function has a special setting for noisy observations with the expense of one more parameter.

As it listed, GPR has a lot of advantages over the conventional regression models. However, GPR can be computational expensive, compared with those models such as OLS models, and the non-linear estimation does make the interpretation difficult.

2.2 Gaussian Process

A Gaussian process is a stochastic process $Y_t,\ t \in \mathbb{R}$, and any realizations of Gaussian process with finite sample size follow joint Gaussian distributions. In other words, on the points $T = \{t_1, \ldots, t_n\}'$ we have the corresponding dependent objective observations $Y = \{y_1, \cdots, y_n\}'$ as a single sample from a multivariate Gaussian distribution $\mathcal{N}_n(\mu(T), V(T))$, where $\mu(T)$ is a n dimension vector and $V(T)$ is a $n \times n$ matrix. In real occasions, T usually represents the real time line.

In order to define the multivariate Gaussian distribution $\mathcal{N}_n(\mu(T), V(T))$, we need to define the mean $\mu(T)$ and the covariance function $V(T)$. According to the theory of Gaussian process, with the knowledge of the independent variable T, the mean and covariance function are both functions of T. For regression purpose, some authors[31] assume that the mean $\mu(X)$ is a zero vector, and attention is focused on choosing an appropriate covariance function. A popular choice is

$$V(t_i, t_j) = \sigma^2 \exp\left[\frac{-(t_i - t_j)^2}{2\lambda^2}\right], \quad i, j = 1, 2, \ldots, n, \tag{2.1}$$

which is a covariance function for any two points. When i is equal to j, we will have the variance function at the point x_i. As we can see here, the variances for different points are assumed to be same. Note that when we are making prediction, the conditional mean is usually not 0 as we assumed above.

Further more, we can use many other formulas for the covariance matrix as long as the formulas have the properties of covariance function. That is the covariance matrix for any

T is positive-definite and symmetric. There are also some examples of covariance function [31] listed in the table 2.1.

Covariance Function	Expression	Parameters
r-Exponential	$\sigma^2 \exp\left(-\frac{\lvert t_i - t_j\rvert^r}{2\lambda^r}\right)$	$\sigma > 0, \lambda > 0, r > 0$
Matérn Class	$\frac{2^{1-v}}{\Gamma(v)}\left(\frac{\sqrt{2v}\lvert t_i - t_j\rvert}{\lambda}\right)^v K_v\left(\frac{\sqrt{2v}\lvert t_i - t_j\rvert}{\lambda}\right)^2$	$v > 0, \lambda > 0$
Rational Quadratic	$\left(1 + \frac{(t_i - t_j)^2}{2\alpha\lambda^2}\right)^{-\alpha}$	$\alpha > 0, \lambda > 0$
Polynomial	$(T \cdot T' + \sigma^2)^p$	$p > 0, \sigma > 0$

Table 2.1: Examples of Covariance Function

Thus (2.1) is just a special situation for r-exponential covariance function when $r = 2$. It only contains two parameters λ and σ. σ is a general control on the magnitude of the covariance and λ the "threshold " parameter, which will be talked about more later.

Considering the form of covariance function (2.1), as t_i and t_j are getting closer (less than λ) $V(t_i, t_j)$ is getting larger. In other words, the closer the points are, the more correlated are the points. On the contrary, if $|t_i - t_j| \to \infty$, there is almost no covariance between these two points. So in r-exponential covariance function, when the power r becomes larger, the covariance between distant points, larger than the threshold parameter λ, will be less. However for the covariance between near points, less than the threshold parameter λ, will be larger.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	0.685	0.662	0.437	0.036	0.002	0.001	0.000
x_2	0.662	0.685	0.543	0.066	0.005	0.001	0.000
x_3	0.437	0.543	0.685	0.230	0.036	0.012	0.000
x_4	0.036	0.066	0.230	0.685	0.436	0.274	0.034
x_5	0.002	0.005	0.036	0.436	0.685	0.632	0.223
x_6	0.001	0.001	0.012	0.274	0.632	0.685	0.376
x_7	0.000	0.000	0.000	0.034	0.223	0.376	0.685

Table 2.2: Example's Covariance Matrix V (three decimal points)

Table 2.2 represents the sample covariance matrix using (2.1) for the seven observations shown in Figure 2.1. All the numbers are rounded into three digits. I have estimated the parameters in (2.1): $\lambda = 0.752$ and $\sigma^2 = 0.685$. We will talk about how to choose parameters in section 2.3. As we expected, the variance are all the same, the covariance are all less than the variance and the matrix is symmetric. The further the two observation, the less the covariance. The covariance between t_1 and t_7 is almost nothing. Noticed that the T in this example is not evenly located, if it is evenly located, we will expect a matrix with the same value for every line parallel to the variance diagonal line.

Actually, because of the covariance function is a function of $|t_i - t_j|,\ i, j = 1, \ldots, n$, we can expect the stationarity of the covariance function. So it makes sense to plot the value of the covariance as a function of $|t_i - t_j|,\ i, j = 1, \ldots, n$ in Figure 2.2. For the points with distance larger than 2, they have very small covariance. When we take a look at the value of the covariance function when $|t_i - t_j| = 0$, we get the estimated σ^2 value is around 0.68.

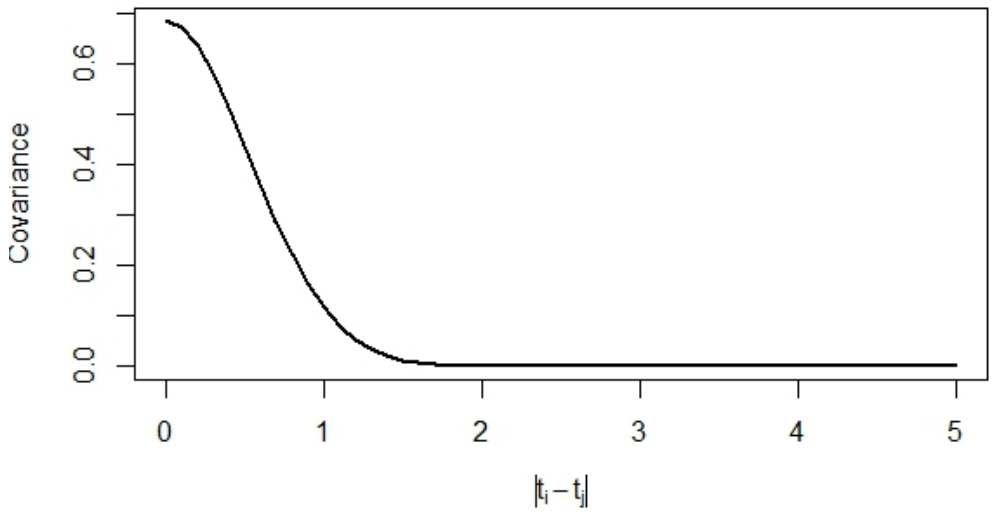

Figure 2.2: Covariance as a function of the distance of the points

Back to the form of the covariance function, in some other situations where the observations are not certain but noisy, we can add error τ^2 to the r-exponential covariance function and redefine the covariance function as

$$V(t_i, t_j) = \sigma^2 \exp\left[\frac{-(t_i - t_j)^r}{2\lambda^r}\right] + \tau^2 \delta(t_i, t_j) \tag{2.2}$$

where $\delta(t_i, t_j)$ is the Kronecker delta function.[3] In the above function, we will expect more variance on every points. However, the covariance between observations will not change. In this form, the parameter τ is coming from the observations' noise. However, there is a shortcoming that, we assume that all the observations are equally noisy, which might be not true in some situations.

In addition, when we are dealing with different types of observations, we can add more components into the covariance function. For example, if there is a periodical pattern in the observation points, we can add

$$\exp\{-u\sin^2[v\pi(t_i - t_j)^r]\} \tag{2.3}$$

to the covariance function (2.1) or (2.2) with three parameters r, u and v. Here v is the parameter which controls the dependence cycle length.

2.3 Gaussian Process Regression

After we have collected the observation points $(t_i, y_i),\ i = 1, 2, \ldots, n$, in practice, it might be a good idea to plot y vs t, which will inform us the choice of covariance function.

Taken together, the observations and the unknown values to be estimated make a finite realization of the Gaussian process. For example, when we are making prediction for one point on t^*, we have a $n+1$ dimensional Gaussian distribution on the unknown value y^* and the observations Y

$$\begin{bmatrix} Y \\ y^* \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \qquad \Sigma = \begin{bmatrix} V & V_*^T \\ V_* & V_{**} \end{bmatrix}, \tag{2.4}$$

where

$$V_* = [V(t_i, t^*)]_{1\times n}, \qquad V_{**} = V(t^*, t^*) \tag{2.5}$$

The covariance matrix is a $(n+1)\times(n+1)$ matrix and the $\mathbf{0}$ represents a $(n+1)\times 1$ vector whose elements are all 0. Then, conditioned on the observations, the estimation point still follows a multi-variable Gaussian distribution.

The wonderful part of the GPR is that distribution of the observations along with the prediction point are defined in (2.4), and the mean is $\mathbf{0}$. However, the conditional

[3] Kronecker delta function: $\delta(t_i, t_j) = 1$ if $i = j$ and o otherwise.

expectation $E(y^*|Y,\theta)$, where $Y = (y_1,\ldots,y_n)'$, θ is the parameters vector, is not 0 but $V_*V^{-1}Y$. This conditional expectation is the estimation we want. Compared with OLS prediction $t^*(T'T)^{-1}T'Y$, $V_*V^{-1}Y$ has similar form and also uses the information about the inverse of some function about set T. In addition, the conditional variances are helpful when we construct the confidence interval. Related conditional distribution deduction will be provided in Appendix B.

In conclusion, the proper conditional distribution for estimation is:

$$y^*|Y,\theta \sim \mathcal{N}(V_*V^{-1}Y,\ V_{**} - V_*V^{-1}V_*') \tag{2.6}$$

If we have k points $T_* = (t_{(1)}, t_{(2)}, \ldots, t_{(k)})$ to estimate, we can just redefine the $V_* = [V(t_{(i)}, t_{(j)})]_{k\times n},\ i = 1,\ldots,n,\ j = 1,\ldots,k$ and $V_{**} = [V(x_{(i)}, x_{(j)})]_{k\times k},\ i,j = 1,\ldots,k$, then substitute them into (2.6). The joint estimation of Y_* corresponding to T_*, $Y_*|Y,\theta$ will just become a multi-variate Gaussian distribution with dimension of k.

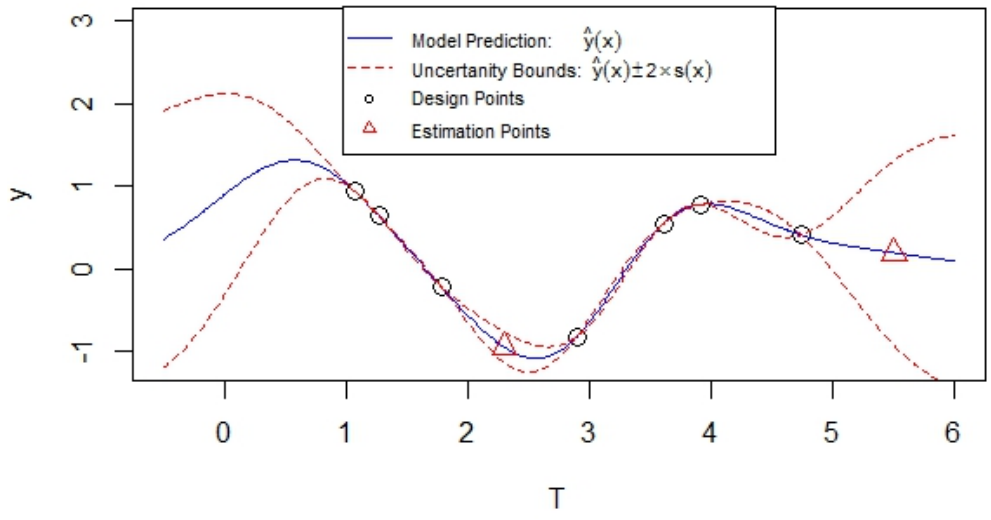

Figure 2.3: GPR Result for the Simple Demonstration

For the example in the section 2.1, the result of GPR when we are using the covariance function (2.1) is shown. As we can see in Figure 2.3, the blue solid line is the prediction line and the two red dashed lines are the 95% confidence interval. Here we can see the heteroscedastic error property clearly. The predictions between the 1_{st} and the 2_{nd}, the 5_{th}

and the 6_{th} points are pretty accurate, because those observations are near for each pair. The longer the distance between the two adjacent observations, the wider the confidence interval. Another property is that the nearer the point to one observation, the narrower the confidence interval. So we have the smooth bow shape confidence interval lines.

The GPR estimation line is much more non-linear than any polynomial models' estimations. In the left end part of the estimation line even turns down as if there is a periodic pattern even if we did not add the periodic term (2.3) into the covariance function. The confidence intervals are wide open in the two ends, which is quite reasonable because the further the points away from observations points the more unsure we are about the estimations.

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
$x^* = 2.3$	V_*	0.180	0.268	0.541	0.502	0.151	0.069	0.004
	V_*V^{-1}	0.874	-1.447	1.178	0.676	-0.679	0.475	-0.069
$x^* = 5.5$	V_*	0.000	0.000	0.000	0.002	0.029	0.074	0.409
	V_*V^{-1}	0.379	-0.573	0.303	-0.429	1.648	-1.846	1.095

Table 2.3: Example's Covariance Matrix V_* on point $t = 2.3$ and $t = 5.5$

In the Table 2.3, it shows when $t^* = 2.3$ and $t^* = 5.5$ in the example, what is the value for V_* and V_*V^{-1}. V_* represents the value of the covariance between estimate points and observation points and V_*V^{-1} shows the coefficients of the observations when we make predictions.

t	$\hat{y}$	$v\hat{a}r$	95% CI
2.3	-0.958	0.009	(-1.144, -0.771)
5.5	0.191	0.327	(-0.930, 1.311)

Table 2.4: Estimation result for the two points

Point $t^* = 2.3$ is right in the middle of the observations and $t^* = 5.5$ is in the end of one side. So we can see point $t^* = 5.5$ is only highly correlated with nearby observation points. The influences of points t_1, t_2 and t_3 on $t^* = 5.5$ are really limited. There is an potential advantage we might be able to take that, when we make prediction, we do not need to take into account of every observation points but only the nearby points, which have enough influences on the prediction point. Together with Figure 2.2, if we have a enough long x domain, we might only consider the influence from the points within $x^* \pm 2$

interval. This might really help to simplify the computation, but we will not look into it further in this thesis. However, point $t^* = 2.3$ is highly correlated with more observations than point $t^* = 5.5$. In other words, to estimate $t^* = 2.3$ takes more information from the observations. So we can see the standard deviation estimation in Table 2.4 for point $t^* = 2.3$ is much less than point $t^* = 5.5$. The confidence interval for point $t^* = 2.3$ is much better than point $t^* = 5.5$.

Then, focused on the coefficient part $V_* V^{-1}$ of the table 2.3. There seems no general rule that the nearer observations have larger absolute coefficient values. However, the distant observations do generally have smaller absolute coefficient values.

2.3.1 Parameter Estimation

Until now, we have the form of the three matrices (2.5). In order to sample from the multi-variable Gaussian distribution (2.4), it is necessary to estimate the parameters $\theta = (\sigma^2, \lambda)$ of the covariance function (2.1). Rarely, we will come across the situation that some parameters are given. Most of the time, we have to find a way to estimate them. The quality of the estimation will affect the quality of the regression directly.

The objective is using the n observations to estimate the parameters of the GPR model. Considering that we have the closed form for the log-likelihood function $\ell(\theta)$, we can use MLE method to choose parameters $\theta = (\sigma^2, \lambda)$ by maximizing $\ell(\theta) \propto \log f(Y|X, \theta)$. We know from (2.4), the multi-variate Gaussian has mean $\mathbf{0}$ and variance matrix V, so the log-likelihood function is

$$\ell(\theta) = -\frac{1}{2} Y^T V^{-1} Y - \frac{1}{2} \log |V| \tag{2.7}$$

where V depends on the parameters $\theta = (\sigma^2, \lambda)$.

By profile likelihood, the 2-d optimization over $\theta = (\sigma^2, \lambda)$ can be simplified into a 1-d optimization over λ as

$$\hat{\sigma}^2(\lambda) = \underset{\sigma^2}{\operatorname{argmax}}\, \ell(\sigma^2, \lambda) = \frac{Y^T M^{-1} Y}{n} \tag{2.8}$$

where

$$M = \left[\exp\left(\frac{-(t_i - t_j)^2}{2\lambda^2} \right) \right]_{n \times n}, \quad i, j = 1, 2, \ldots, n \tag{2.9}$$

The profile likelihood is defined as:

$$\ell_{profile}(\lambda) = \ell(\lambda, \hat{\sigma}^2(\lambda)) = -\frac{n}{2} - \frac{1}{2}\left(\log|M| + n\log\left|\frac{Y^T M^{-1} Y}{n}\right|\right) \tag{2.10}$$

Upon maximizing $\ell_{profile}(\lambda)$, the values $\hat{\lambda}$ and $\hat{\sigma}^2(\hat{\lambda})$ will be the precise MLE estimators which maximize the original log likelihood function $\ell(\theta)$. Details will be provided in Appendix A. In other words, we can turn this multi-parameters optimization problem into an one parameter optimization. However, this formula can only be used when the covariance function is defined as in (2.1).

Computationally, GPR parameters estimation can be tricky. The problem is mainly on the evaluations of the likelihood function. Term $-0.5\log|V|$ in the likelihood function is described as the "complexity penalty term" [33]. The MLE for estimating the parameters is not that simple in some situations. There are some methods suggested, for example, the genetic algorithm [30] and the multiple starting locations search method [21]. In addition, close points will easily generate a near-singular situation for the matrix, which makes numerical computation of inverse and determinant of matrix difficult. To be specific, this near singular situation will generate some very small negative eigenvalues, which is supposed to be very small positive numbers, and leads to the failure of inverse. So when we run an optimize algorithm with bad starting values or too wide estimate intervals, there will be an error. In order to overcome this problem, "nugget" is introduced into the computation [14] [37] procedure, and advanced work to improve the accuracy of the estimation was made by Ranjan (2011) [30]. Even a further improvement is given in Butler (2013) [5] and a Matlab package 'GPMfit' is developed.

In addition, this profile likelihood method can also be used in some other covariance functions forms as long as we can separate the covariance function into two part as

$$k(x_i, x_j) = f(\theta_1)g(\theta_2, \ldots, \theta_m) \tag{2.11}$$

where we have m parameters. Then we can obtain the estimation of the θ_1 given $\theta_2, \ldots, \theta_m$ by profile likelihood method. This property can be useful when we discuss the Multi-responses GPR later.

Until now, we outlined the GPR methodology with a simple example. GPR's popularity is due to its several advantages over the other linear and non-linear regression models. Compared with most of the other models, GPR is a more powerful method [32] in the respective of estimation by generating highly-nonlinear estimations. GPR's estimations have heteroscedastic errors: the points nearer the observations, the narrower the confi-

dence intervals. In most of the cases, heteroscedastic error is better than the same error everywhere as we will get by polynomial regression. In addition, the number of parameters are controlled under GPR and GPR is so simple and easy to implement [22]. Compared with neural network models, which are usually contains too many parameters and followed by problems like easily overfitting and too complex to implement, GPR is a better choice. However, GPR do have some disadvantage as well. GPR lacks the power of interpretation. It is quite easy for us to tell the trend from a linear model, but the non-linear path from GPR can be a problem to interpret. Moreover, GPR is computationally more expensive than polynomial regression. Because we have to run optimization algorithm to estimate the parameters instead of has close form likelihood derivative.

2.4 Multi-response GPR

Multi-response GPR, also known as multi-task or multi-output GPR, helps to deal with observations containing multiple response levels. One of the first few work was done by Cressie (1993) [7]. In his famous book about spatial analysis, Cressie studied the bi-output and three-output situations.

Except the simple one response cases[4], there are also some situations that the response is multi-dimensional. If different responses are independent, then we can just treat those multi-response cases as several one-response cases. So we can just apply the simple GPR to every response level. However, the usual situations are that there are correlations between those responses. In order to fit these situations, we should modify the GPR methodology.

Assuming that we have N observations lie at points $t_1, \ldots, t_N$, each of which has M responses. Then we define a matrix for the responses,

$$Y = (Y_1, \ldots, Y_N)' = \begin{bmatrix} y_{11}, y_{12}, \cdots, y_{1M} \\ \vdots \qquad \ddots \qquad \vdots \\ y_{N1}, y_{N2}, \cdots, y_{NM} \end{bmatrix}, \tag{2.12}$$

where y_{ij} means the j_{th} response of the i_{th} observation and every vector in the matrix is one response. Y can be viewed as a sample from a multi-variate Gaussian distribution by letting

$$\text{vec}(Y) \sim \mathcal{N}(\mathbf{0}, C) \tag{2.13}$$

[4] "One response" here means a response vector corresponding to a vector of time line.

The choice of covariance function is not as simple as for one-response GPR [13]. The problem is that we have more observation values than time points and the covariance function for one-response GPR is not working here. The potential increase of the number of parameters is huge, considering the complexity of multi-response framework. In the work of Bonilla, Chai and Williams (2008) [3] and the work of Boyle and Frean [4], they explored some forms of covariance function for multi-response GPR. This is a popular choice:

$$\operatorname{cov}(y_{p,i}, y_{q,j}) = \sigma_{p,q}^2 \exp\left[\frac{-|t_i - t_j|^r}{2\lambda^r}\right] \tag{2.14}$$

which is covariance function for response $y_{p,i}$ and response $y_{q,j}$.

This covariance function assumes that the correlation between observations can be separated into two parts. One is for the observations with different t values and the other is for the inter-observation covariance. The covariance among the different responses for all the observations are assumed to be the same. The covariance among different observations are the same as defined in r-exponential covariance function. While these are indeed very strong assumptions, they simplify the covariance function and offer enough interpretation power we shall see later.

This covariance function (2.14) is based on the r-exponential covariance function. If the circumstance needs, we can always change the second part of (2.14) into any covariance function in Table 2.1.

Here we shall define another two matrices

$$\Sigma = (\sigma_{i,j})_{M\times M} \tag{2.15}$$

where M is the response dimension, $0 \leq i, j \leq M$, and

$$V = \left[\exp\left(\frac{-|t_i - t_j|^r}{2\lambda^r}\right)\right]_{N\times N} \tag{2.16}$$

where N is the observation points, $0 \leq i, j \leq N$.

V matrix represents the covariance between the different observations and the Σ matrix represents the covariance among different responses in each observation. After we define

these two matrices, we can introduce the kronecker product of these two matrices,

$$C = \Sigma \otimes V := \begin{pmatrix} \sigma_{1,1}V & \cdots & \sigma_{1,M}V \\ \vdots & \ddots & \vdots \\ \sigma_{M,1}V & \cdots & \sigma_{M,M}V \end{pmatrix} \tag{2.17}$$

which is a $NM \times NM$ matrix corresponds to the $M \times N$ covariance matrix C we defined in (2.13). By analogy to the one-response case, multi-response GPR proceeds as follows:

1. We maximize the log likelihood function:

$$\ell(\theta) = -\frac{1}{2}\text{vec}(Y)^T C^{-1}\text{vec}(Y) - \frac{1}{2}\log|C| \tag{2.18}$$

   where θ includes λ, r and all the N^2 elements in Σ. It seems that there are $N^2 + 2$ parameters in total and it will be difficult to maximize the likelihood function over so many parameters. However, there are some advantages we can take by applying profile likelihood and the properties of kronecker product into the optimization, which we will look into in the next section.

2. The estimation y^* for one new point t^* will be

$$y^*|Y, \theta \sim \mathcal{N}(C_* C^{-1}\text{vec}(Y),\ C_{**} - C_* C_{-1} C_*^T) \tag{2.19}$$

   where

$$C_* = \Sigma \otimes \left[\exp\left(\frac{-|t^* - t_i|^r}{2\lambda^r}\right)\right]_{N\times 1}$$

   which is a $NM \times M$ matrix and

$$C_{**} = \Sigma \otimes \exp\left(\frac{-|t^* - t^*|^r}{2\lambda^r}\right)$$

   which is a $M \times M$ matrix. The result is that $y^*|Y, \theta$ is still a M-dimension Gaussian distribution.

2.4.1 Computational Advantage

If we consider 1000 data points, and the observations may have two dimensions. A 2000×2000 matrix is the covariance matrix, and we have to compute the determinant and inverse

of this matrix when we estimate the parameters θ. Assuming we are given a power value r, it still leaves us 5 parameters r, $\lambda, \sigma_{1,1}, \sigma_{1,2}, \sigma_{2,1}, \sigma_{2,2}$[5] to estimate. There is obviously no closed form for the optimization procedure[6], because the likelihood function involving the determinant of a matrix. Considering the number of observations is large, it can be challenging to optimize over so many parameters. Even if we can get the estimations right, it will take a long time to optimize.

To simplify this problem, We apply the profile likelihood method. In the condition that we have already known λ and r, we can actually calculate matrix V. Then perform a Cholesky decomposition to matrix V and we can get $V = U'U$, where U is a lower triangular matrix with real and positive diagonal entries. According to the property of covariance matrix, V should be invertible, then we shall have a matrix U^{-1} as the inverse matrix of U.

If we are given

$$Z = U^{-1}Y \tag{2.20}$$

where Z is a $N \times M$ matrix, then according to the profile MLE, we have

$$\hat{\Sigma} = \frac{1}{N}Z'Z = \frac{1}{N}Y'V^{-1}Y = (\hat{\sigma}_{i,j})_{M\times M} \tag{2.21}$$

Now, we can substitute the profile MLE result into the likelihood function (2.18) together with the property of Kronecker Product and get

$$\ell_{profile}(\theta) = -\frac{N}{2}\sum_{i,j=1}^{M}(\hat{\sigma}^*_{i,j}\sigma_{i,j}) - \frac{1}{2}(M\log|V| + N\log\left|\frac{Y'V^{-1}Y}{N}\right|) \tag{2.22}$$

where $\hat{\sigma}^*_{i,j}$ is the (i,j) element of $\hat{\Sigma}^{-1}$ and $Y_i = (y_{1i}, y_{2i}, \ldots, y_{Ni})$. Using this formula, we can narrow down the high-dimension optimization into a two-dimension optimization which is much simpler. Before using profile likelihood, we face $NM \times NM$ matrix inverse. But now we only need to compute $N \times N$ matrix inverse.

All the deduction part of the profile likelihood can be found in Appendix A.

Until now, we talked about some basic ideas about GPR, and we shall move on to some examples.

[5]The covariance matrix is symmetric, so $\sigma_{i,j} = \sigma_{j,i}$

[6]Taking partial derivative does not have a close form

Chapter 3

Simulation Study with FitzHugh-Nagumo Model

3.1 FitzHugh-Nagumo Model

The FitzHugh-Nagumo (FHN) model is a set of ordinary differential equations (ODE). This model's solution process has some features of the biological neural excitement system[42].

The FHN model is defined by two variables. V is the voltage variable and R is the recovery variable:

$$\frac{\partial V}{\partial t} = c(V - \frac{V^3}{3} + R) \tag{3.1}$$

$$\frac{\partial R}{\partial t} = \frac{1}{c}(V - a + bR) \tag{3.2}$$

with three parameters (a,b,c). If we have a initial conditions (V_0, R_0), we have a outcome for $t > 0$ as plotted in Figure 3.1.

The FHN model is composed of a linear (3.2) equation and a nonlinear (3.1) equation. As the V values grows, we may have a sharp curve for V plot and we may have a more smooth curve for R plot. Specificly, if we have small time value $t > 0$, the main influence will be cV instead of $cV^3/3$ so we will see an almost linear line; around the point $\sqrt{3}$, as t grows, $cV^3/3$ will quickly take over the control power leaves a sharp turn until R changed correspondingly and the system will reach another balance. Although V value changes sharply, R equation is only a linear function of V value. Compared with the V equation, which is a cubic function of V value, R curve will change less violently than V curve. So

is it shown in Figure 3.1 with the parameters $(a,\ b,\ c) = (0.6,\ 0.4,\ 2.5)$ and the initial conditions are $(V_0,\ R_0) = (0,\ 0)$.

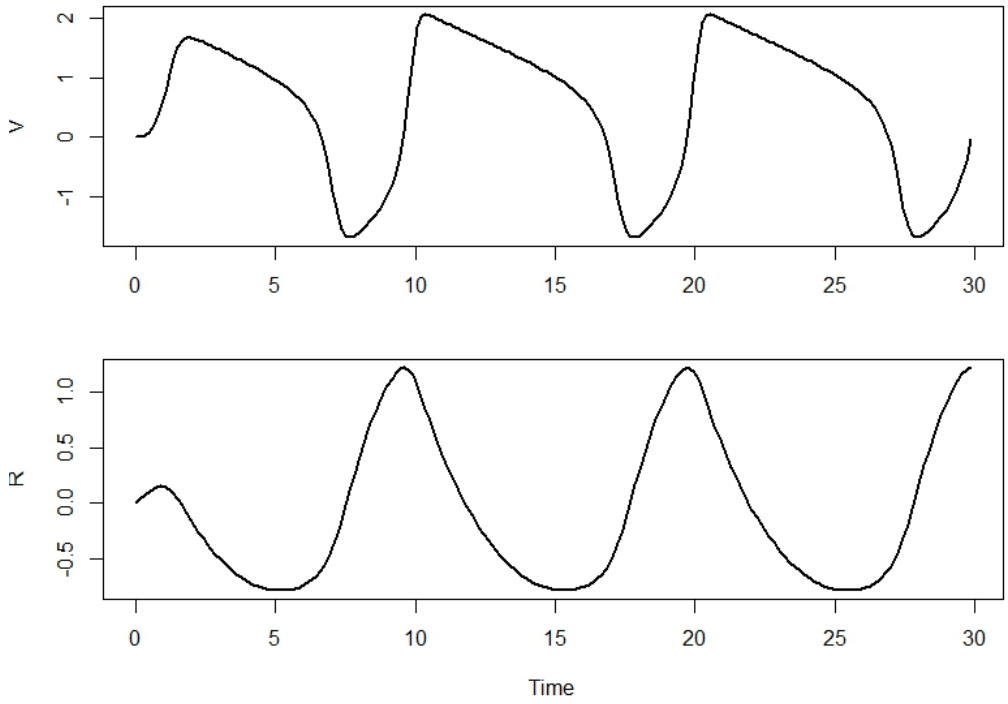

Figure 3.1: FHN model plot with parameters a=0.6, b=0.4, c=2.5 and initial value $V_0 = 0, R_0 = 0$

Estimation of the parameters for the FHN model is a notoriously challenging task[29]. There will exist many local optima for those dynamic systems modelling by Esposito and Floudas[11] and many other problems. So ordinary optimization algorithm might not as useful as we thought in this problem. [29] gives a review of several eligible methods and propose a solution based on a modification of data smoothing methods.

Here, we shall attempt to use GPR to recover the curves for V and R without estimating the parameters a, b and c. GPR has a non-linear estimation curve. With enough observations, GPR is expected to recover the curves nicely.

3.2 Simulation Study I

Now we want to see how good GPR can recover the curve if we only know limited points of the curve. In order to discuss some properties of GPR, 30 randomly sampled time points from the interval $[0, 30]$ are used. Those are, as we can see in the Figure 3.5, the blue dots on the black curve. Here, we focus on the V curve first, we will get to R curve later.

To explore the parameter r in the r-exponential covariance function, we use

$$V(t_i, t_j) = \sigma^2 \exp\left[\frac{-|t_i - t_j|^r}{2\lambda^r}\right], \quad i, j = 1, 2, \ldots, n \tag{3.3}$$

as the covariance function in this chapter.

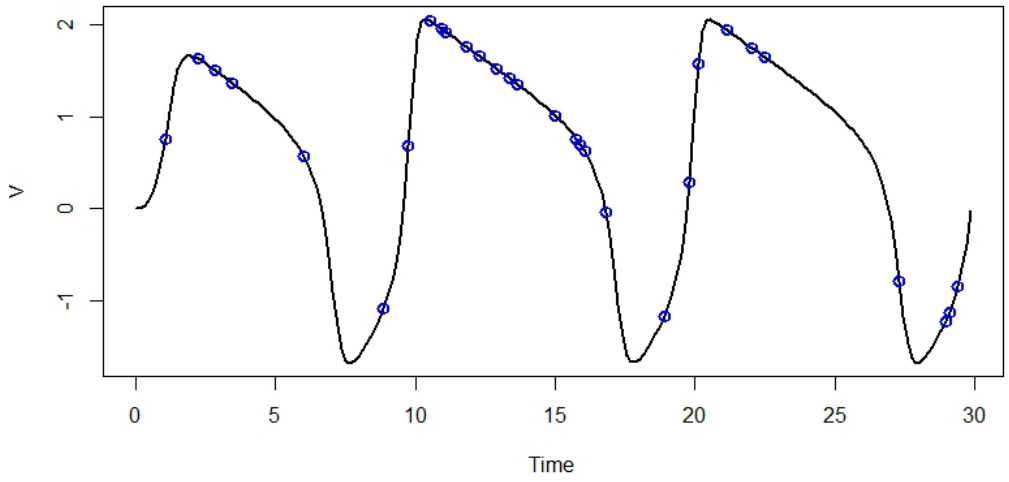

Figure 3.2: 30 random sample points on FitzHugh-Nagumo model plot

Here we are using covariance function (3.3), we have three parameters $\theta = (\sigma^2, \lambda, r)$. By analogy to the profile likelihood as we discussed in chapter 2, we can have a profile MLE $\hat{\sigma}^2 = \hat{\sigma}^2(\lambda, r)$. The following proceeds as

$$\ell_{profile}(\lambda, r) = \ell(\lambda, r, \hat{\sigma}^2(\lambda, r)) = -\frac{n}{2} - \frac{1}{2}\left(\log|M| + n\log\left|\frac{Y^T M^{-1} Y}{n}\right|\right) \tag{3.4}$$

where

$$M = \left[\exp\left(\frac{-(t_i - t_j)^r}{2\lambda^r}\right)\right]_{n\times n}, \quad i, j = 1, 2, \ldots, n \tag{3.5}$$

So we only need to estimate two parameters (λ, r). In order to choose the best parameters estimations, we have a contour plot of the likelihood function value over λ and r. Too large r value will decrease the covariance and λ, as the threshold parameter, should have a reasonable value corresponding to the length of the sample interval [0,30]. So we have r in the range of [0,2] and λ is in the range of [0,5].

Judging from Figure 3.3, the local best estimation should be around point $(\lambda, r) = (1.2, 1.75)$ or point $(\lambda, r) = (0.85, 1.9)$. To run a optimize algorithm with these points as starting value, we get the MLE $(\hat{\lambda}, \hat{\sigma}^2, \hat{r}) = (0.837, 1.374, 1.997)$.

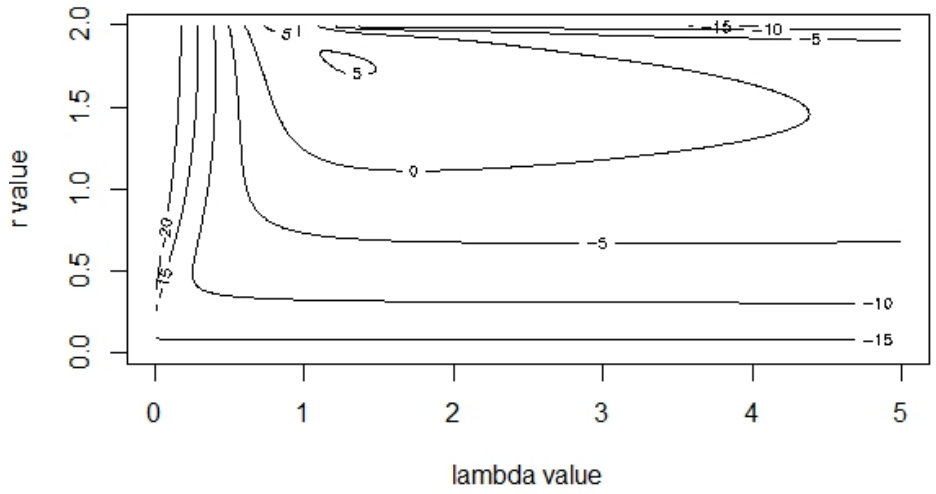

Figure 3.3: GPR parameters estimation contour plot

Figure 3.4 displays the GPR predictions. Almost all the black solid true curve stays inside the 95% confidence interval. It is clear that the nearer the observation points, the narrower the confidence interval between the two observations. The nearer the prediction point to an observation, the narrower the confidence interval for it. So we have some bow shape confidence interval cruves as we can see. Between the time interval 10 to 15, the estimation is extremely good with small confidence intervals and precise prediction line because the observations are dense here. Between time 2 to 4 and 22 to 30, we have reasonably dense observations, the estimations are also satisfying.

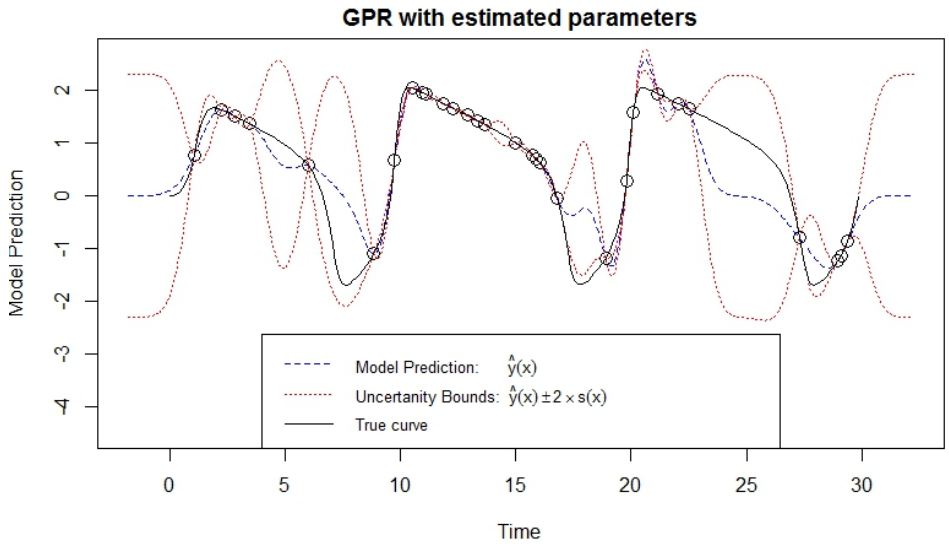

Figure 3.4: GPR for the first set of 30 sample points with r=1.8

However, we still have to notice that there are some defects. For the sharp turn at around time 18, there is no sample point on the turn point so the true line is outside the confidence interval. The similar turn point around time 8, the true curve stays inside the confidence interval, however around time 8 the confidence interval are wider than time 18. Around the time interval 22 to 27 in the third hump, the adjacent observations are too far away. The true curve is convex a little bit but the estimation curve is concave. Although the true curve in this part is absolutely inside the 95% confidence interval, the confidence intervals are really wide. Noticed the same part in the second hump, the observations actually cover the part the third hump does not cover. So it means the simple covariance function (3.3) cannot capture the periodic property of this model. We might want to try to add periodic term later.

In all, GPR makes the prediction in a different way and the prediction curve is not like any fixed form regression model. If it does not make too much impression in the simple example in chapter 2, here we can see GPR prediction curve is so flexible. It is so much more than some simple line between two observations. GPR predicts confidence interval according to the estimated point's distance to the nearest observation, which is quite reasonable unlike every prediction has the same variance everywhere.

3.3 Simulation Study II

To further explore the properties of GPR, r is a important parameter in the covariance function. We will check that how different r can affect the prediction results. In addition, it also makes sense to make predictions with different sets of samples. So we will exam four different sets of randomly sampled samples with GPR. Further more, as we mentioned in the last section, we are interested about the result by adding a periodic term into the covariance function. Finally, FHN model is a two variables model, multi-response GPR can also be applied here.

3.3.1 GPR with different r

Now we check how the different r value in covariance function (3.3) will affect the results of GPR. We still use the sample from last section, sample 1[1], and use (3.3) as covariance function. Different from last section, we use 4 different r values (1.0, 1.3, 1.7, 2) to carry out GPR and we got four plots in the left side of Figure 3.5. There are several interesting results:

1. The estimation curves for these four r values have different curvatures in details. It is clearest to compare the part of plots between time 6 to 9. When $r = 1.7$, the estimation curve shows clear trend of going down, then a round smooth turn up to the next observation. However, for $r = 1.3$ and $r = 1.0$ the estimation curves are almost straight during the time interval and then angularly turn up. Judging from time period 16 to 19 and 22 to 27, I will conclude that with larger r value, we will expect more severe curvatures on the estimation cruves.

2. Greater r values have better confidence interval during those small time intervals, but not necessarily better during those big time intervals.

   Let us focus on the curves between time 12 and 17, the observations are relatively dense here. The GPR prediction with $r = 2$ achieve almost no variation. As the r value become smaller, the wider the confidence interval.

   To explore what is the reason, we check the covariance function:

$$V(t_i, t_j) = \sigma^2 \exp\left[\frac{-|t_i - t_j|^r}{2\lambda^r}\right], \quad i, j = 1, 2, \ldots, n \tag{3.6}$$

[1]We will expect different sets samples in the next part

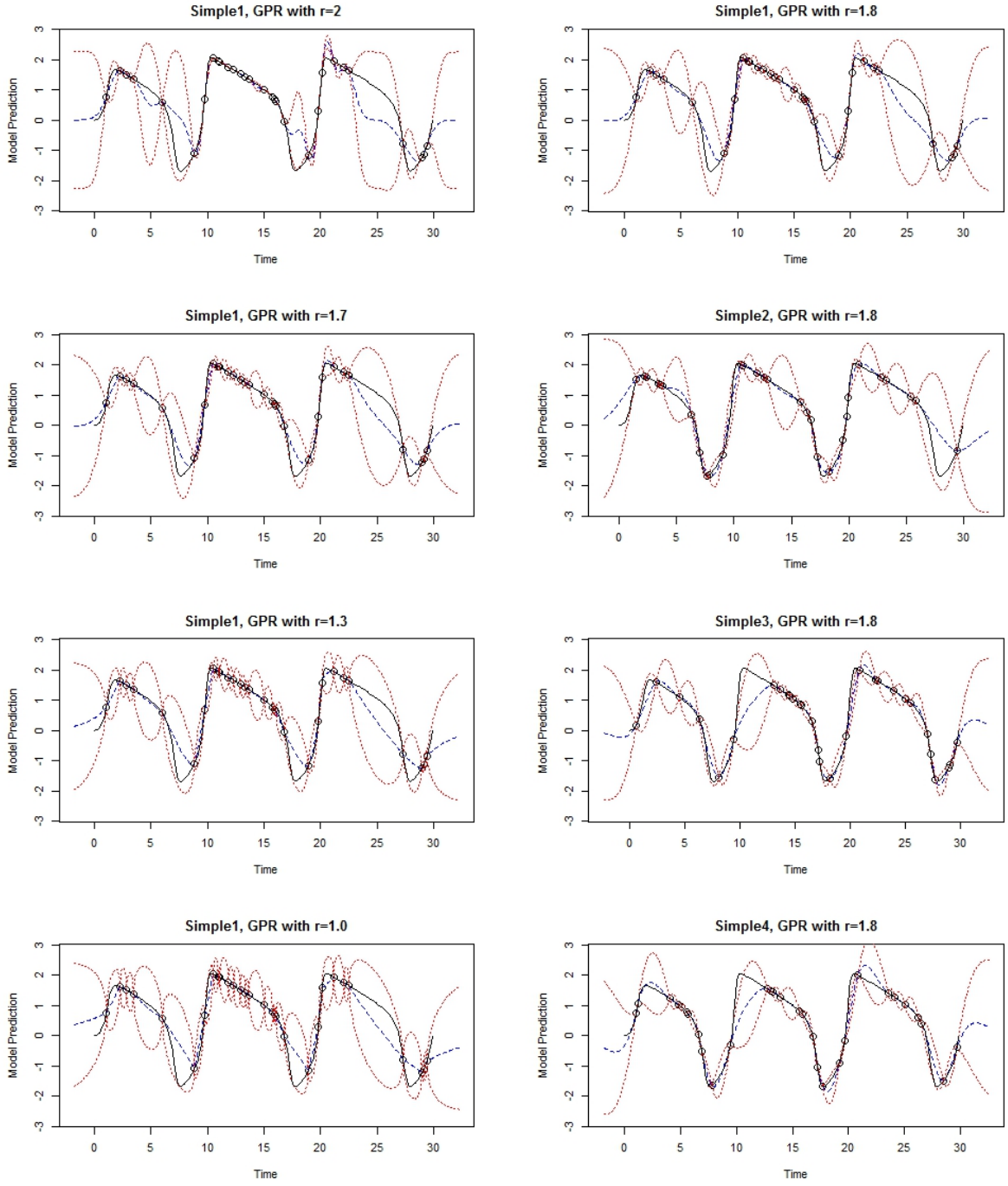

Figure 3.5: GPR results with different samples and different r values

When we have $|t_i - t_j| > \lambda$, as r value increases, the covariance will decrease. On the contrary, when we have $|t_i - t_j| < \lambda$, as r value increases, the covariance will increase as well. λ is a linear control of the covariance function and r is a exponential control of the covariance function.

	$r = 1.0$	$r = 1.3$	$r = 1.7$	$r = 2.0$
$\hat{\lambda}$	1.885	1.628	1.371	0.799

Table 3.1: Different λ values corresponding to the different r values

In Table 3.1, we can check the different MLE values for λ when we using different r values. All of them are less than 0.5 and λ is decreasing as r increases. During the time interval 12 to 17, all the observations are apart less than time 0.5. We will expect more correlation on the time points in this time interval when we using larger r value. More correlations lead to small variations on those points.

However, if we focus on the time 22 to 27, we cannot tell the confidence intervals are that different for different r values. It is because in the middle of this interval, some points are less than λ, but more are larger than λ. The conditions are not the same, so we cannot directly apply the rule above here.

According to the rule we deduct above, for small time intervals (time interval less than any 2λs), we will have more correlation between the middle points and the two observations with larger r value. So we will have narrower confidence interval with larger r value. However, for those adjacent observations with too big time interval (larger than any 2λs), the right middle time point will have less standard deviation when r is smaller.

Obs.	Time-int.	Mid-Point	r value	$\hat{y}$	$\hat{s.d.}$
x_{11}, x_{12}	0.45	21.6	1.0	1.7042	0.2875
			1.3	1.7065	0.1983
			1.7	1.7062	0.1133
			2.0	1.7011	0.0029
x_{26}, x_{27}	4.80	24.9	1.0	0.3515	0.8833
			1.3	0.3193	0.9269
			1.7	0.2996	1.0841
			2.0	-0.0034	1.1491

Table 3.2: Analysis the different r value

let's take a look at the result in Table 3.2, the result shows when the time interval is 0.45, the estimated s.d. for the middle point decreases as r value grows and when the time interval is 4.8, the estimated s.d. is largest when r is the largest.

It should be pointed out that when we are applying GPR to the real life problem, the time is usually calculated by year. So the time interval usually will be less than λ, which leads to a conclusion that the larger r we choose, the better confidence interval we will get if corresponding λ values are similar.

3. Smaller r values posses rounder confidence intervals curves, and greater r values yield confidence intervals more like two bows. In other words, for the points that nearest the observations, those with larger r value will have narrower confidence intervals.

   To find out what is the reason for this, we have to look back at the covariance function. As we discussed in the second result, those points that close to the observations will have $|t_i - t_j| < \lambda$. As a result, we will expect narrow confidence intervals with larger r value.

3.3.2 GPR with different sample points

Now we sample four different sets of random samples with equal sample size of 30. We will apply covariance function (3.3) with $r = 1.8$ to all of these samples. Then we have the plots for the true curve, prediction curve and confidence intervals in Figure 3.5 and we have those results:

1. No matter how to choose those 30 sample points, the GPR prediction 95% confidence interval usually cover the true curve. All the four samples GPR results' red smaller dashed curves contain almost all the true curve, although some observations are quite distant with each other. However, we cannot help but notice that in the third plot, the true curve goes outside the 95% confidence interval both around time 10 and 20. For time 10, it is because the time interval between the two observations are too apart. The estimation curve is terrible compared with the true line too. However, for time 20, it is because the time interval between the observations are too small but the vertical distance is huge. Although the estimation curve is not too bad but the turn point is just too sharp for GPR to predict.

2. To make good predictions about the sharp turn, an observation near the turn point is crucial. Let's take a look at the turn points around time 8 and 18 in sample 1 and sample 2 plots. The estimation curve for sample 2 around those points are

much better than the estimation curve for sample 1. It is clear that sample 2 has observations near the turn points, but sample 1 has not. It is not surprising to see that the confidence intervals are better for sample 2 around time 8 and 18.

3. The estimated confidence interval width depend mainly on the adjacent observations' time interval. Here is a Table 3.3, and the results are from sample 3.

Observations	Time Interval	Middle Point	$\hat{var}$	$\hat{s.d.}$
x_{17}, x_{18}	1.5	18.90	0.0675	0.2599
x_{21}, x_{22}	1.5	23.25	0.0623	0.2496
x_{18}, x_{19}	1.2	20.25	0.0440	0.2098
x_{27}, x_{28}	1.2	28.35	0.0363	0.1905

Table 3.3: Analysis the estimated s.d.

First of all, the longer the time interval, the greater the middle point's estimated variance. There are two sets of time intervals in Table 3.3. Apparently, time interval 1.5 has estimated s.d. around 0.25, but time interval 1.2 has estimated s.d. around 0.20.

Secondly, the locations of the time intervals are not so relevant to the estimated variance for the middle points. For example, the vertical distance between x_{17} and x_{18} is much more than x_{21} and x_{22}. We figure that the quality of estimation will be better for the time interval between x_{21} and x_{22}, but actually the estimated variances are almost the same. The situation is the same for observations x_{18}, x_{19} and x_{27}, x_{28}. So we say the influences of the locations of the time intervals to the quality of estimation are limited.

3.3.3 Periodic Term

Until now, we have talked a lot about the properties of GPR with (3.3) as covariance function. Let us attempt another covariance function with periodic term and see how well are the predictions. As we can see in Figure 3.1, V value plot shows a clear sign of periodicity. Section 2.2 actually mentioned that we can add a term to the original covariance function to help to model the periodicity. So now we introduce the combined covariance function with periodic term

$$k(x_i, x_j) = \sigma_f^2 \exp\left[\frac{-|x_i - x_j|^r}{2\lambda^r}\right] + \sigma_p^2 \exp\{-u \sin^2[v\pi|x_i - x_j|]\}, \quad i, j = 1, 2, \ldots, n. \quad (3.7)$$

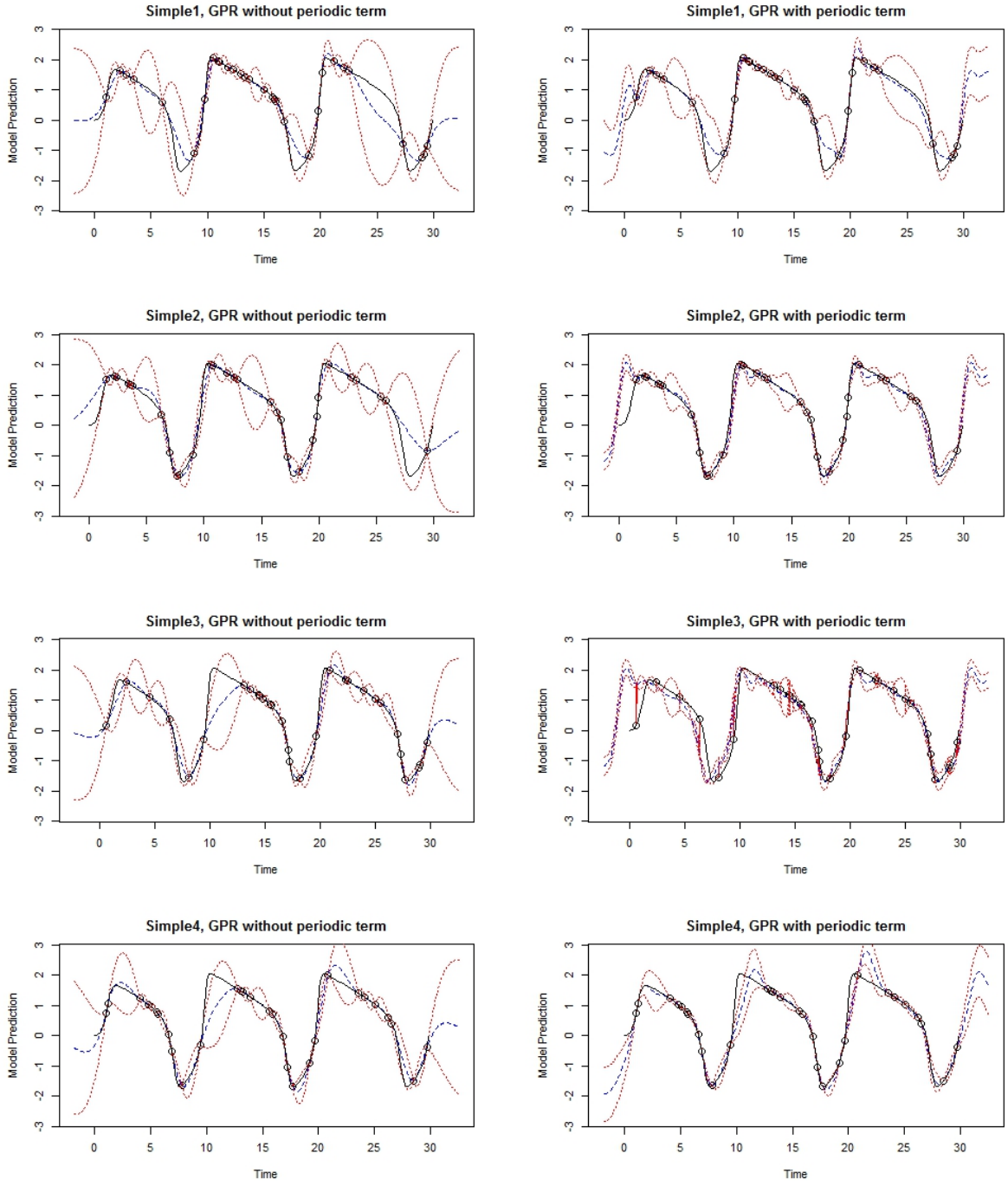

Figure 3.6: Comparison of with or without periodic term in GPR with r=1.8

The results are shown in Figure 3.6. In this example we are still using the same four samples as we used before.

GPR with covariance function (3.3) cannot achieve a good estimation curve during the big gaps as we can see, for example, sample 1 estimation around the time interval 23 to 27 and sample 2 estimation around time interval 26 to 29. On the contrary, as we can see in the right side plot, GPR with covariance function (3.7) estimates almost perfectly during the time interval of 23 to 27, at least significantly better than the left side plot. It is because we have a lot of information in the first and the second humps. Especially in the second hump, the observations are almost evenly distributed around it. However, still using sample 1, for the valley parts of time 6 to 8 or 16 to 17, no matter adding the periodic term into the covariance or not, we cannot get good estimations. The reason is because sample 1 does not have observations at the bottom of the valley. While sample 2 have enough sample points around the valley parts during the first and the second valleys, so we can get a pretty good estimation curve using (3.7) during the third valley even if we do not have any observations there. It is interesting that GPR generate a strange estimation path using Sample 4. Although we have one observation on the top of the third hump, but the estimation for this part is still not so good. I think it is the property of large r value. GPR with large r value will generate severely curve estimations path. If we have another observation a little right to the single observation on the top, we might get a perfect estimation path.

In the other perspective, GPR using (3.7) as covariance function has better confidence intervals during the time 23 to 27 and the two tails in sample 1 plot. With periodic term, the estimation curves have clear trends of periodicity on both tails, instead of no clear trends of without periodic term.

To further explore the covariance function (3.7), we shall give another set of assumptions. In the real situation, we might have evenly distributed observations, e.g. daily data or monthly data, then we might want to make a prediction about the the future period. In Figure 3.7, the first plot shows the result of GPR make prediction about the future period. The estimation curve is quite good and the confidence intervals does not get worse as time goes on. It is interesting that the first prediction hump is not as good as the second prediction hump. It might be due to that we have an observation in the first hump so the estimations are drawn to this observation.

If we give several scattered observations in the future, then we have the second plot in Figure 3.7. As we can see, compared with the plot without those extra points, the second plot has slightly better estimation curve and the confidence intervals are much better. This is a example for the potential situation that we have some missing data. When we have

more dense observation and less missing data, we will have even better estimations.

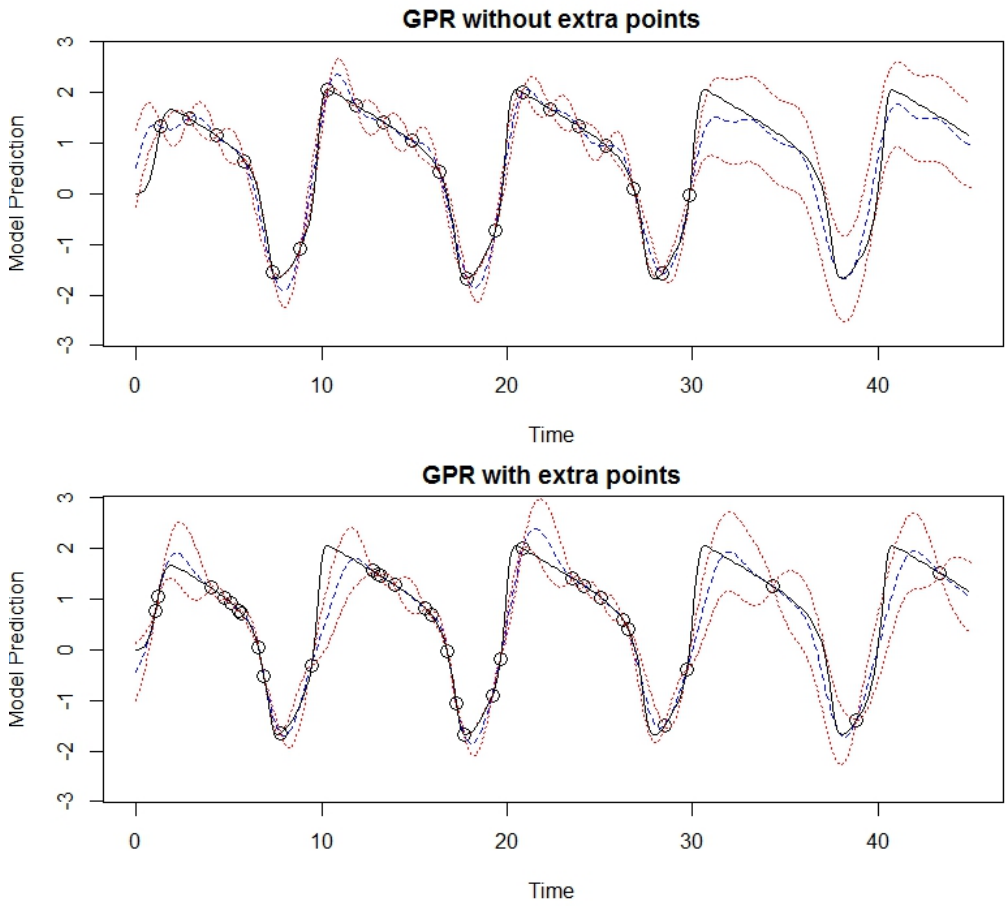

Figure 3.7: GPR with evenly distributed observations using periodic term

3.3.4 Bi-variable GPR on FHN model

Judging from the FHN ODEs, V value and R value is related. Further more, there is a inverse correlation as we can see in Figure 3.1. We do have data for pairs of V and R. So it might be a good choice to use the Multi-responses GPR. Here we still use sample 1 for example.

First, we want to estimate the parameters. In the same parameters domain r in $[0, 2]$ and λ in $[0, 5]$, we get the contour plot. The pattern is similar to the other contour plot

in the previous part. The best estimation is around point $(r, \lambda) = (1.8, 1.8)$. We might as well choose a nearby starting point for the optimize algorithm and computer the MLE's of the parameters.

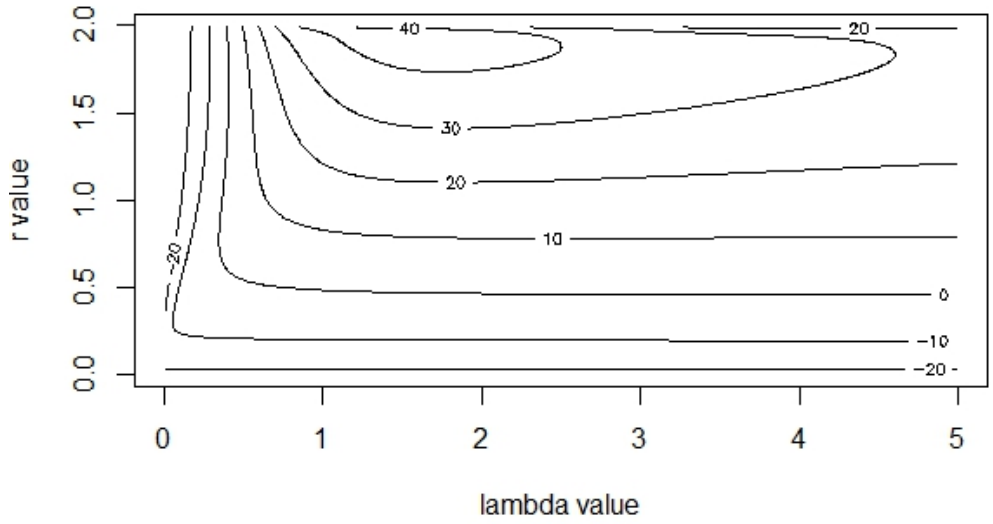

Figure 3.8: Multi-responses GPR parameters estimation contour plot

The result of the parameter estimations are

$$\hat{\Sigma} = \begin{pmatrix} 5.222 & -0.2130 \\ -0.2130 & 0.3186 \end{pmatrix}, \qquad \hat{\lambda} = 1.681, \qquad \hat{r} = 1.920,$$

While the result shows that the estimation of the correlation is not so large, which will definitely related to the quality of the estimation. To calculate actually correlation, we find that the true correlation between V samples and R samples is -0.347, which is larger than the result from multi-response GPR.

In order to compare the result from the multi-response GPR and the simple GPR, Figure 3.9 is plotted. The two simple one-rsponse GPR estimations use the same $r = 1.920$ as the estimated value in the multi-response GPR. First of all, not all the confidence intervals for multi-response GPR are better than the simple GPR. For example, for the points in V plot around time 25, simple GPR has better confidence interval. However, the left tail part in R plots, multi-response GPR is better. Secondly, the estimation curves from multi-response GPR are usually better than the results from simple GPR. It is clear

that during the valley parts in V plots, the multi-response GPR estimation curve is much closer to the true curve. Still in the V plots, during time 23 to 27, multi-response GPR estimation curve gives much better estimation.

To find out what is the reason that multi-response GPR gives bad confidence interval around those big gaps, we should check the λ values. The λ for V is 0.943 and for R is 1.917. It is clear that V's λ is much smaller than multi-response GPR. As we discussed in the previous part, λ is the threshold parameter that determine the critical value of the distance between observations. One point t^* estimation with different λ, the larger the λ, the smaller the value $(|t^* - t|/\lambda)^r$, the larger the covariance function. So it does make sense that the confidence interval around 25 is bad from multi-response GPR and better from simple GPR. Simple GPR's λ value fot R part is larger than multi-response GPR, so we can see that the confidence interval for the two tails in simple GPR is worse than multi-respose GPR.

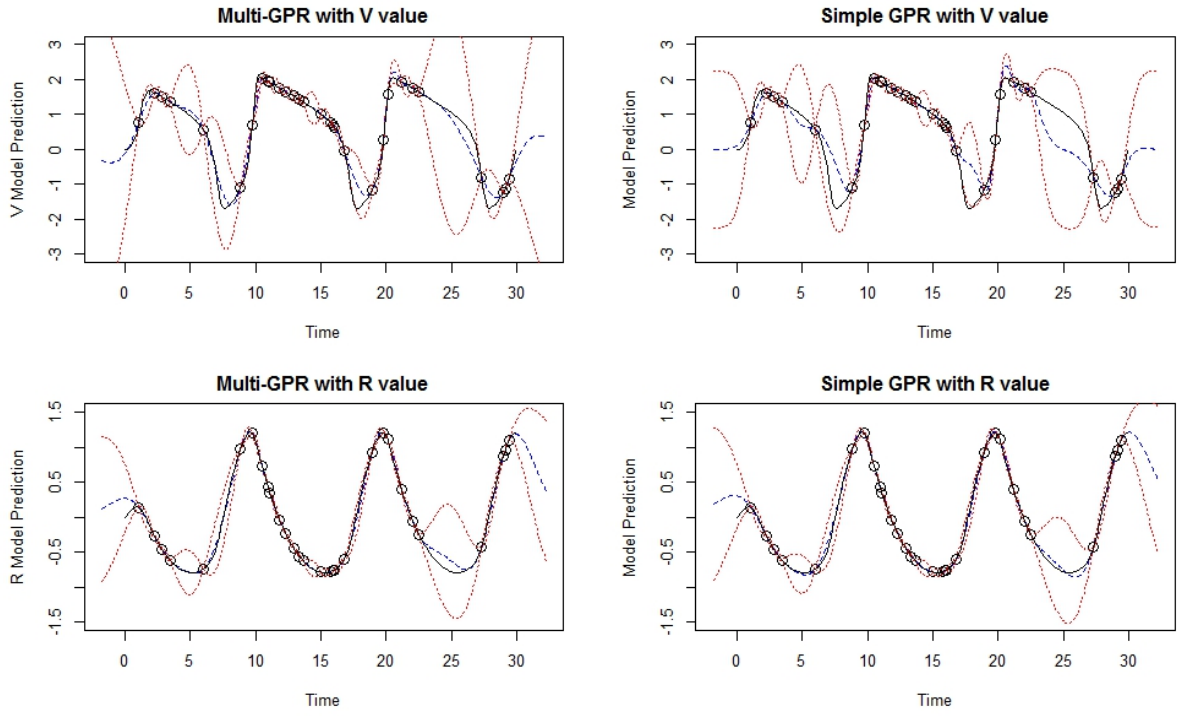

Figure 3.9: Multi-reponses GPR result compares with Single-variable GPR result

Chapter 4

Inference for Stochastic Differential Equations

4.1 Missing Data Problem

Real time is a continuous scale and many data, e.g. financial data, are based on the real time scale. Such data are often modeled using stochastic differential equations (SDEs). An SDE for continuous stochastic process X_t is written as

$$\mathrm{d}X_t = \mu(X_t, \theta)\mathrm{d}t + \sigma(X_t, \theta)\mathrm{d}B_t \tag{4.1}$$

where θ is a set of parameters for this SDE, and B_t is the Brownian motion.

Although the SDE is specified in continuous time, we can only record discrete observations $X = (X_0, X_1, \ldots, X_n)$ physically. For simplicity, let us assume a constant inter-observation time Δt.

Given the discrete observations, the likelihood function for the parameters is

$$L(\theta|X) \propto \prod_{i=1}^{N} p_{\Delta t}(X_i|X_{i-1}, \theta), \tag{4.2}$$

where $p_{\Delta t}(X_i|X_{i-1}, \theta)$ is the transition density of X_t which, by construction, is a Markov process. However, it is extremely rare that the transition density of a SDE model can be expressible in closed form.

In order to solve this inference problem, a popular approach is to discretize the continuous-time equation using the Euler approximation:

$$X_{t+\Delta t} \approx X_t + \mu(X_t, \theta)\Delta t + \sigma(X_t, \theta)\Delta B_t \tag{4.3}$$

where $\Delta B_t = B_{t+\Delta t} - B_t \sim \mathcal{N}(0, \Delta t)$. Thus the transition densities can be approximated by Gaussian distributions,

$$X_{t+\Delta t}|X_t, \theta \sim \mathcal{N}(X_t + \mu(X_t)\Delta t, \sigma^2(X_t)\Delta t). \tag{4.4}$$

The accuracy of this Euler approximation increases as $\Delta t \to 0$. If the actual inter-observation time Δt is large, the Euler approximation will significantly bias inferential results [12].

If we are not satisfied with the data resolution of the observations X, we can treat the midpoints between observations as missing data. That is, let $X_{miss} = (X_{0.5}, X_{1.5}, \ldots, X_{N-0.5})$ be the midpoints between the observation points X. For example, if we have the same time interval Δt between observations, then $X_{i-0.5}$ is the middle point at time $(i - 0, 5)\Delta t$ between observation X_{i-1} and X_i at time $(i-1)\Delta t$ and $i\Delta t$.

By Markov property, the joint density of the complete data is

$$\begin{aligned} p(X, X_{miss}|\theta) &= p(X_0|\theta) \prod_{i=1}^{N} p(X_{i-0.5}|X_{i-1}, X_{i-1.5}, \ldots, X_0, \theta) \times p(X_i|X_{i-0.5}, X_{i-1}, \ldots, X_0, \theta) \\ &= p(X_0|\theta) \prod_{i=1}^{N} p(X_{i-0.5}|X_{i-1}, \theta) \times p(X_i|X_{i-0.5}, \theta) \end{aligned}$$

By introducing the missing data like this, the data resolution is doubled. Now every transition density can be better approximated by the Euler discretization. That is, from a Bayesian perspective, the original Euler posterior is

$$p_{\Delta t}(\theta|X) \propto \pi(\theta) \cdot \prod_{i=1}^{N} p_{\Delta t}(X_i|X_{i-1}, \theta) \tag{4.5}$$

by adding the missing data, we have

$$p_{\Delta t/2}(\theta|X) \propto \pi(\theta) \int \prod_{i=1}^{N} \left[p_{\Delta t/2}(X_i|X_{i-0.5}, \theta) p_{\Delta t/2}(X_{i-0.5}|X_{i-1}, \theta) \right] dX_{miss}. \tag{4.6}$$

When we replace all the true transition densities in the above two equations with Gaussian densities by Euler approximation, $p_{\Delta t/2}(\theta|X)$ is a better approximation then $p_{\Delta t}(\theta|X)$. By analogy, we can add more missing data between the observations. For example, we can add three missing values between each pair of observations X_{i-1} and X_i at time point $t_i + \Delta t/4$, $t_i + \Delta t/2$ and $t_i + 3\Delta t/4$. Thus we can have $p_{\Delta t/4}(\theta|X)$, which is even a better approximation to the real parameter likelihood than $p_{\Delta t/2}(\theta|X)$, due to Euler approximation is better with less time interval. Assuming that the integrals in (4.6) and all the other $p_{\Delta t/k}(\theta|X)$ can be evaluated, then we can get an approximation that converges to the true SDE posterior $p(\theta|X)$ as $k \to \infty$.

However, the integrals cannot be computed analytically, so the idea is to apply "simulated likelihood method" [15] and sample from

$$p(X_{miss}, \theta|X) \propto \pi(\theta)L(\theta|X, X_{miss}). \tag{4.7}$$

If we can have samples $(\theta^{(1)}, Y_m^{(1)}), (\theta^{(2)}, Y_m^{(2)}), \ldots, (\theta^{(N)}, Y_m^{(N)})$ from this target density, then we can just ignore Y_m and $(\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(N)})$ are the samples we needed to make inference. This sampling procedure is usually done by Markov Chain Monte Carlo (MCMC) method. The detail of implement of MCMC can be found in the work of Eraker (2001) [10] and the work of Beskos (2008) [2]. However, our idea is to use GPR to create a Monte Carlo sampling algorithm, which is easier than MCMC.

4.2 GPR-Based Importance Sampler

Monte Carlo integration is methodology to solve a integration problem with an approximated expectation as a sample average.

$$E(X) = \int f(x)p(x)dx \approx \frac{1}{M}\sum_{i=1}^{M} f(x_i), \tag{4.8}$$

where $p(x)$ is the density function and $x_i, i = 1, \ldots, M$ are i.i.d. samples from $p(x)$.

Based on Monte Carlo integration, an importance sampling is a very fast way of obtaining samples from a target density $p(x)$ when we can only sample from some auxiliary proposal density $q(x)$. It means to solve a integration problem as

$$\int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx \approx \frac{1}{M}\sum_{i=1}^{M} f(x_i)\frac{p(x_i)}{q(x_i)} \tag{4.9}$$

where $x_i, i = 1, \ldots, M$ are i.i.d. samples from $q(x)$.

Thus the importance sampler proceeds as follows:

1. Obtain i.i.d. samples from the proposal distribution $x_1, \ldots, x_M \sim q(x)$.
2. For each x_i, calculate weights $w_i = p(x_i)/q(x_i), i = 1, \ldots, M$.
3. Calculate normalized weights $r_i = w_i / \sum_{j=1}^{M} w_j$, in case we only know the proportional value $p(x)$ instead of the true target density.
4. Sample N values among $x_1, \ldots, x_M$ with replacement and with normalize weights $r_1, \ldots, r_M$. Those new N samples are recognized as samples from $p(x)$.

In this case, the random variable of interest is (θ, X_{miss}) and the target density is $p(\theta, X_{miss}|X)$. When there is no closed form for the transition density, we will use Euler approximation and get

$$\hat{p}(\theta, X_{miss}|X) \propto \pi(\theta)\hat{L}(\theta|X, X_{miss}), \tag{4.10}$$

where, using one missing data scenario as example, we have approximation:

$$\hat{L}(\theta|X, X_{miss}) \propto \prod_{i=1}^{N} \hat{p}(X_{i-0.5}|X_{i-1}, \theta) \times \hat{p}(X_i|X_{i-0.5}, \theta) \tag{4.11}$$

where every approximate transition densities are all approximated by Gaussian distribution using Euler approximation.

If we can find a good proposal distribution, then we can use the procedure described above and sample posterior θ. However, the difficult for importance sampler is obtaining a good proposal distribution $q(\theta, X_{miss})$. A good proposal distribution requires a density has even coverage over the target density and it should be easy to sample from. To solve this question, a lot of work has been done, such as Kou (2012) [17] was using a method of parallel sampling between different Euler approximations. But they used Gibbs sampling, this thesis proposes a importance sampler method.

In this thesis, we construct proposals in a different way by first decomposing the joint posterior distribution

$$p(\theta, X_{miss}|X) = p(\theta|X, X_{miss}) \times p(X_{miss}|X). \tag{4.12}$$

where $p(\theta|X, X_{miss})$ is the conditional parameter distribution and $p(X_{miss}|X)$ is the conditional missing data distribution.

In many financial applications, the conditional parameter distribution is analytically available, which means that we can have the density and can sample from it. But the conditional missing data distribution is usually not. We propose GPR prediction can serve as a good approximation, thus importance sampler proposal is

$$q(\theta, X_{miss}) = p(\theta|X, X_{miss}) \times p_{GPR}(X_{miss}|X). \tag{4.13}$$

Until now, we can analytically write down the target density and proposal density, which are both easily to sample from. Noticed that we need to properly normalize some of the results, they are written in proportion forms.

4.3 CIR Model

As a famous mathematical finance model, Cox-Ingersoll-Ross model (CIR model) imitates the behavior of interest rates. It named after John Cox, Jonathan Ingersoll and Stephen Ross (1985) [6], who introduced it. CIR model assumes that only market risk affects the interest rate, and it follows a stochastic differential equation defined as

$$\mathrm{d}r_t = a(b - r_t)\mathrm{d}t + \sigma\sqrt{r_t}\mathrm{d}B_t \tag{4.14}$$

where $a > 0,\ b > 0,\ \sigma > 0,\ \theta = (a, b, \sigma),\ B_t$ is the Brownian Motion represents the influence of the random market risk.

CIR model is a stochastic process with the drift part $a(b - r_t)$ and the Brownian part imitates the random market risk. Judging from the form of the drift part of the model, it is a mean-riveting around b:

$$E[r_t|r_0] = \exp(-at)r_0 + (1 - \exp(-at))b. \tag{4.15}$$

The larger the interest rate r_t, the more the random market influences the interest rate. When r_t goes to 0, the drift term is positive and the influence of the random term is limited. It is customary to impose the condition $2ab \leq \sigma^2$, ensures all the interest rate r_t is larger than 0. The other two parameters a and σ control the scale of the changes in drift and random risk respectively.

The CIR model is one of the only SDEs admitting a closed form for its transition

density. The transition density is a noncentral chi-square distribution [6]:

$$p(X_{t+\Delta t}|X_t,\theta) = ce^{-u-v}(\frac{v}{u})^{p/2} I_v\left(2(uv)^{1/2}\right), \tag{4.16}$$

[1] where

$$p = \frac{2ab}{\sigma^2} - 1, \quad c = \frac{2a}{\sigma^2(1-\exp(-a\Delta t)}, \quad u = cX_t \exp(-a\Delta t), \quad v = cX_{t+\Delta t}$$

Since the transition density is known, there is no need to resort to the Euler approximation and missing data. We use the CIR model as an analytic test scenario to benchmark GPR's ability to impute the missing data between actual observations.

To simulate data from CIR model, we choose parameters as $a = 3.76$, $b = 0.179$ and $\sigma = 0.401$. Time interval is set to be $1/252$ to represent the daily data and data size is set to be 1000.

To simplify the problem, we will use the missing data scenario when only one missing data $X_{i-0.5}, i = 1,\ldots,N$ is between each pair of observations as we defined before. Using the analytic transition density formula, a single missing point $X_{i-0.5},\ i = 1,\ldots,n$ has the conditional density

$$p(X_{i-0.5}|X,\theta) \propto p(X_{i-0.5}|X_{i-1},\theta) \times p(X_i|X_{i-0.5},\theta). \tag{4.17}$$

Like this, we can evaluate the conditional distribution of all those intermediate points. Then we will compare the true density with Euler approximation and two GPR predictions. Euler approximation can be achieved the same as (4.17) by replacing all the true transition densities with approximated Gaussian distribution. Including Euler approximation results is just a simple demonstration about the quality of Euler approximation. As far as the two GPRs, we just use the points prediction results.

Here, we will use the r-exponential as the covariance function. The first GPR approach is to optimize over two parameters (r,λ). This approach, using the MLE r value, chooses parameters through the information from observations (GPRr). However, GPR with r-exponential covariance function has Markov property when $r = 1$, which is an unique property that all the other r value does not possess. Actually, Ornstein-Uhlenbeck (OU) process is one of the few Gaussian Processes with close form SDE, and OU process has a covariance function exactly the same as r-exponential function with $r = 1$. However, all the other gaussian processes with r-exponential covariance function do not have a close

[1] $I_a(x)$ is the modified Bessel function

form SDE to describe it. By the property of SDE, we know the Gaussian process with $r = 1$ exponential covariance function has the Markov property. Considering that CIR model's Markov property, we use it as a second GPR approach (GPR1).

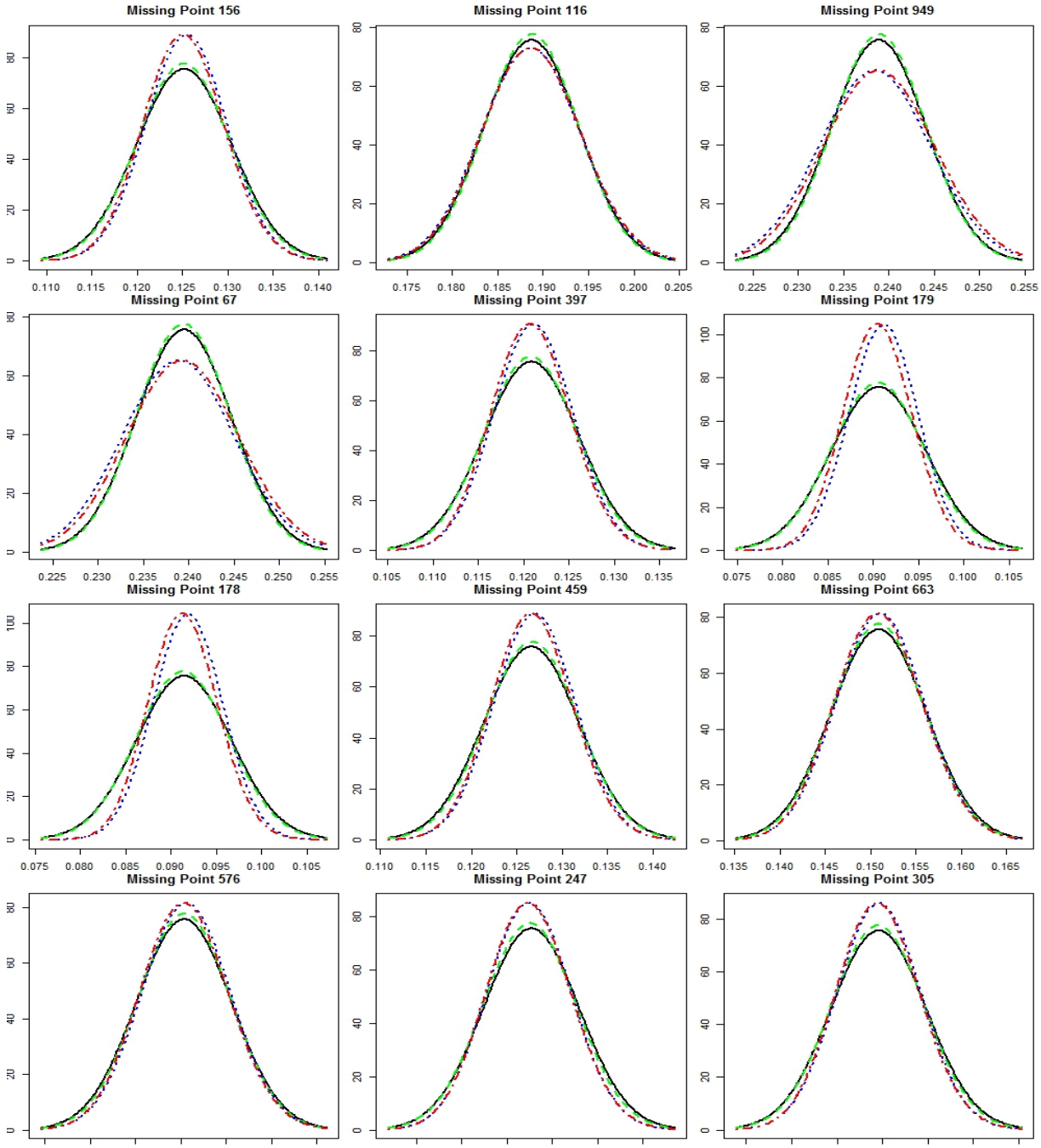

Figure 4.1: Daily CIR Missing Point Likelihoods Comparison.
Black solid line: GPRr approximation; Green dashed line: GPR1 approximation;
Red dotdash line: True SDE likelihood; Blue dotted line: Euler approximation.

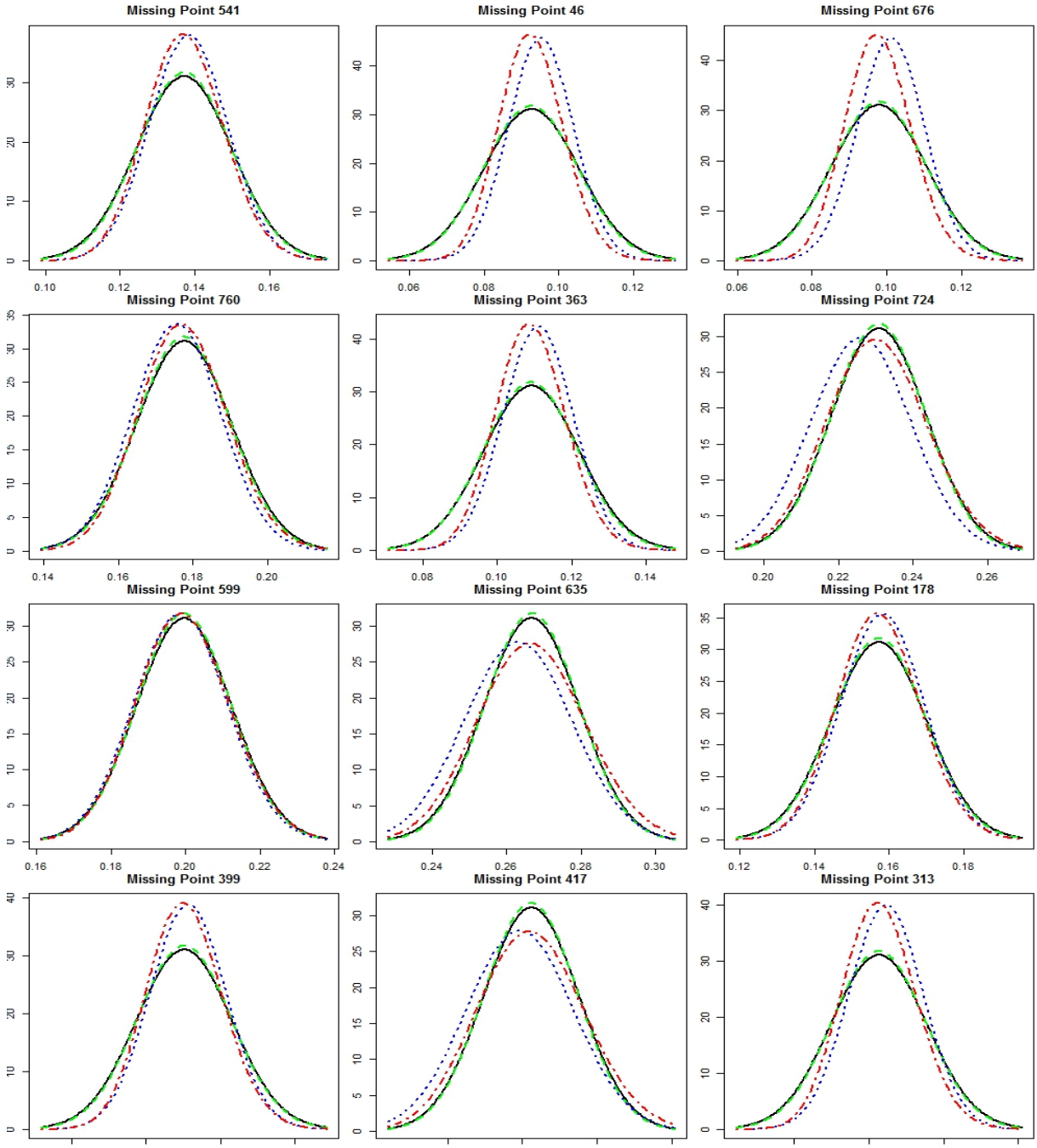

Figure 4.2: Weekly CIR Missing Point Likelihoods Comparison.
Black solid line: GPRr approximation; Green dashed line: GPR1 approximation;
Red dotdash line: True SDE likelihood; Blue dotted line: Euler approximation.

GPRr's MLE for parameters (r, λ) are $(0.962, 13.395)$. As we can see the estimation of r is less than 1. We have GPR1's λ estimation is 10.503. Combining the result from chapter 3, GPR1 will have smaller variance value, because the λ value are all larger than time interval $1/252$. In other words, GPR with $r = 1$ will have likelihood curve sharper. Noticed the GPRr estimated r value is quite close to 1, it shows the data also suggest the markov property of the model.

In Figure 4.1, we have the comparison plots among true densities, Euler approximation densities, GPRr approximation and GPR1 approximation. The corresponding colors and line type can be found in the caption of the figure. Every plot is for a single point and all of these 12 missing points are generated randomly.

As we expected, the likelihoods for the true densities and the Euler approximation densities are almost the same. Except that some of them have slightly different mean locations, as we mentioned that Euler approximation can be biased as time interval grows larger. Green dashed curve, which is GPR1, is always a little shaper than black solid curve, which is GPRr. As we discussed in chapter 2, larger r value yields smaller variance. But the r values of the two GPR models are not too different, so the results of the estimation actually do not differ that much. Judging from the 12 plots, in more than 60% of the points, GPR performs really well and the likelihoods are almost the same as the true densities. Even the worse situations at missing point 178 is not that bad. Except in missing point 949, GPRs densities are wider than or almost equal to the true densities in all the other plots. In (4.13), we need $p(X_{miss}|X)$, which is the GPR estimation, instead of $p(X_{miss}|X, \theta)$, which is the true density and Euler density. In other words, GPR estimations do not need the information about the parameters but true density and Euler density need. Using less information, GPRs' larger standard deviations fit the theoretical deduction.

For the daily data, we can conclude that GPR approximation is quite good. However, the weekly data might be challenging for the GPR method, because the time interval is larger. Figure 4.2 shows the weekly data comparison plots. The result is that GPRs still perform quite well. Focus on the mean locations, apparently GPR estimations do better than the Euler approximation. The standard deviations of GPRs as we discussed are still reasonably good.

So far, GPR performs well on estimations of the missing data problem based on the CIR model, and it shows great potential to achieve as good estimations based on a more complicated SDE. This SDE may have no close form for the transition density. We shall move on the next section for the Heston model's inference.

4.4 Heston Model

Heston Model is a stochastic volatility model with a pair of SDEs, one of which describes the volatility of an asset by a simple process based on Brownian motion and the other represents the price of the asset based on its volatility. In another word, the volatility of the asset is not a simple Brownian motion but another process.

Here is one form of the SDEs of Heston model by performing a transformation $Y_t = (X_t, Z_t)' = (\log(S_t), 2V_t^{1/2})'$ base on the original variables (S_t, V_t) of Heston model:

$$\begin{aligned} \mathrm{d}X_t &= (\alpha - \frac{1}{8}Z_t^2)\mathrm{d}t + \frac{1}{2}Z_t \mathrm{d}B_{X_t} \\ \mathrm{d}Z_t &= (\beta/Z_t - \frac{1}{2}\gamma Z_t)\mathrm{d}t + \sigma \mathrm{d}B_{Z_t} \\ \rho &= \mathrm{cor}(B_{X_t}, B_{Z_t}) \end{aligned} \tag{4.18}$$

which will be a nicer form to use when we come across the computations.

4.4.1 Estimation Results

The basic idea here is to see how good is GPR fitting the model. If GPR turns out to be a good fit, then we will look into the possibility of performing importance sampling based on GPR.

As mentioned before, many MCMC methods have been proposed to sample from the joint posterior of the parameters and missing data. However, it is computational expensive. If GPR densities on missing points can achieve nice results, then computationally speaking, importance sampling can be much more efficient than the MCMC approach. As there are two variables in Heston model and clearly those two variables are correlated, this might be a good case to use multi-responses GPR on these variables (X_t, Z_t).

Thanks to the code provided by my supervisor M. Lysy, I am able to generate samples from a standard Gibbs sampler for the Euler posteriors at different resolutions [20]. Resolution k means that we discretize the adjacent observations equally into 2^k parts and impute $2^k - 1$ missing points. The higher the k resolution we can achieve better results from Euler approximation. For example, if we are using $k = 1$, we are sampling for only one intermediate points between adjacent observations.

To simulate data from Heston Model, we use parameters $\alpha = 1$, $\gamma = 5$, $\beta = 0.82$, $\sigma = 0.6$ and $\rho = -0.8$, and we give it a reasonable large time interval $5/252$ to represent

weekly data. We simulate 50 observations leaving 49 points as missing data. In addition, we will sample from MCMC method by iterating 500, 000 times, which should be enough to make converging estimation. Resolution 0 to 2 will be run and the result data is stored. The same as CIR model data, we will concentrate on the one intermediate missing data between observations scenario. So resolution 1 data keeps all the samples for the missing data we need.

The reason we choose weekly data is according to the posterior estimation for the parameters with this time interval. Resolution 1 is different from resolution 0 but close to resolution 2 as we can see in the first plot of Figure 4.3. This means that resolution 1 is a sufficient approximation for the true SDE, instead of using resolution 2. However, this information will not be known in advance. If we choose a smaller time interval, resolution 1 will not be significantly different from resolution 0. This means that missing data imputation is not needed, because the original observations have already secured a good parameters likelihood approximation.

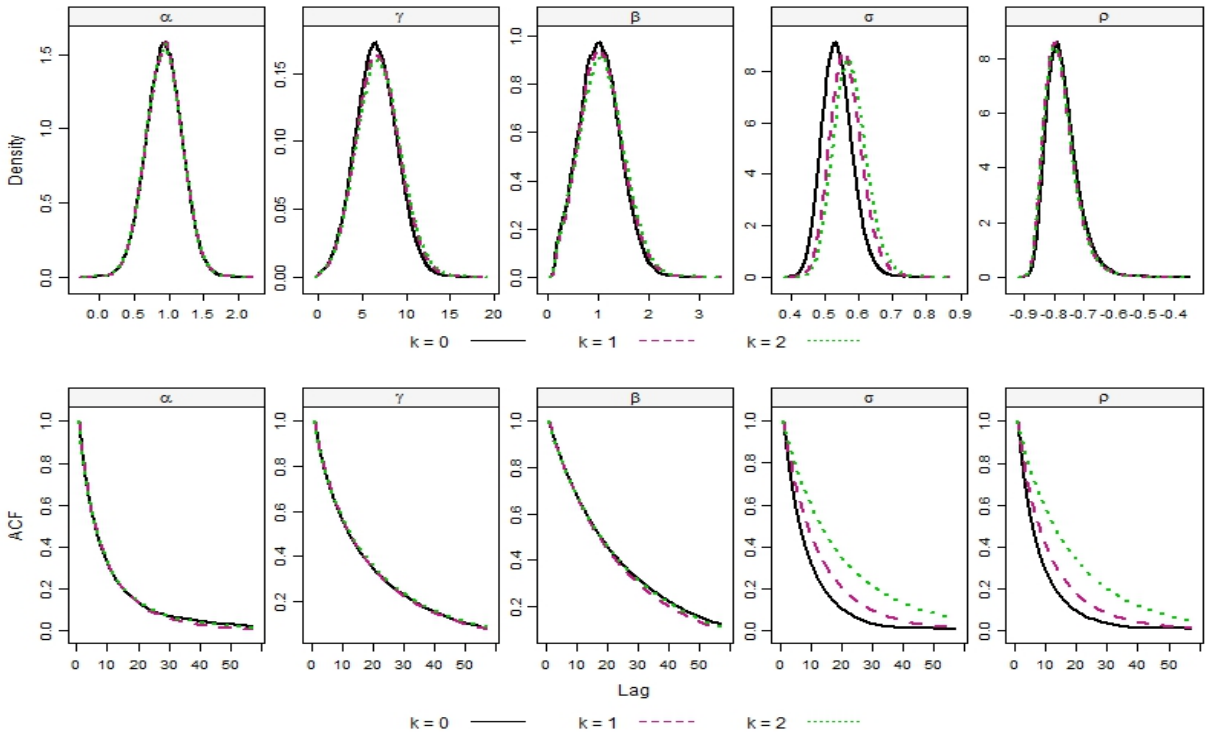

Figure 4.3: MCMC Posterior Parameters Inference with 500,000 Iterations

MCMC takes too long to calculate and the results are correlated. The second plot in Figure 4.3 shows the autocorrelation (ACF) for samples of each parameters. It shows the heavy correlation between MCMC samples. For all five parameters, the samples within 10 lags have more than 30% correlation. But if we use importance sampling, all the samples are independent. As k grows, the autocorrelation problem becomes worse. Here, resolution 1 is our focus, because we are essentially trying to compare the results from GPR based importance sampling with it.

$500,000$ iterations using MCMC take more than 70 seconds on my computer using C++ code. However, draw $1e7$ samples from 99 dimension Gaussian distribution takes about the same time using C++ code in the same computational settings. Indeed, importance sampling from Gaussian process is much faster than MCMC method. We might want to consider changing the iteration numbers for MCMC, but too less iterations just cannot reach convergence. Figure 4.4 shows the posterior parameter inference. As we can see that compared with Figure 4.3, likelihood for γ, β, α haven't totally converged yet.

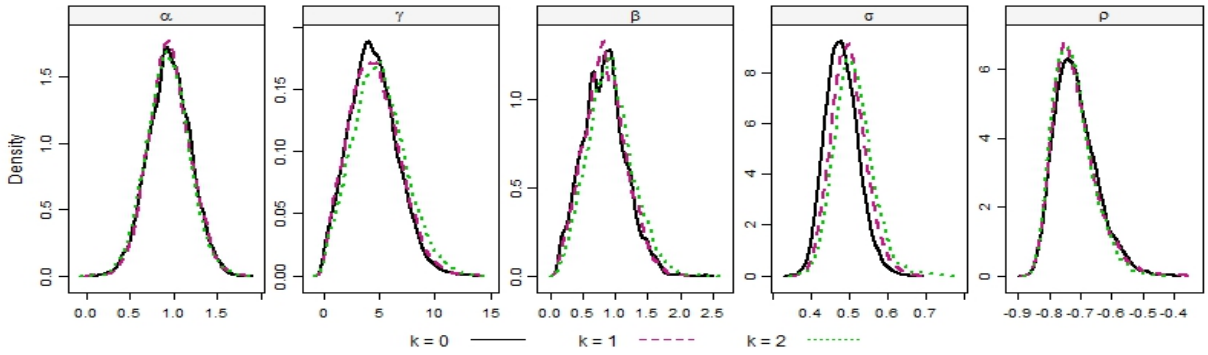

Figure 4.4: MCMC Posterior Parameters Inference with 20,000 Iterations

Before we apply the importance sampler based on GPR, let us first take a look at the quality of the GPR estimations. We have the sample data from MCMC, so it makes sense to compare the distribution quantile from GPR with numerical quantile from MCMC data. MCMC sample data is used as the true samples in this chapter. The confidence intervals from GPR are constructed by $mean \pm 1.96 \times sd$, while the confidence intervals from MCMC sample data is just the result of the sample quantiles. We still use GPRr and GPR1 here for the same reason we mentioned in the last section. In Figure 4.5, we can see the comparison for these three method: sample quantile and two GPR methods. In order to facilitate the visual display, results are presented after re-centering the MCMC true missing data to have mean equal to 0.

Two GPR methods both are quite good with the mean estimations, as we can see that both black solid and red dashed line stays around 0. The estimated r value is 1.313, which is much larger than 1, so that GPRr has much narrower confidence intervals. Remarkably, GPR1 produces an excellent estimate of the missing data in Z_t. However, GPRr consistently undercovers the true missing data distribution's support. This will have very negative consequences for importance sampling as we shall soon see. For the X_t component, both GPR approximations have considerable difficulty capturing the true missing data density. As we know that the volatility of X_t is partially determined by the value of Z_t. Smaller Z_t gives us smaller X_t volatility. Maybe we should introduce the value Z_t into the GPR methodology to meet the requirements of the Heston model, but the basic settings of the GPR we are using now just simply cannot imitate the behavior of X_t. Luckily enough, we have confidence intervals from GPR with $r = 1$ cover almost all the sample quantile confidence interval, so we want to proceed to the importance sampling part using GPR with $r = 1$.

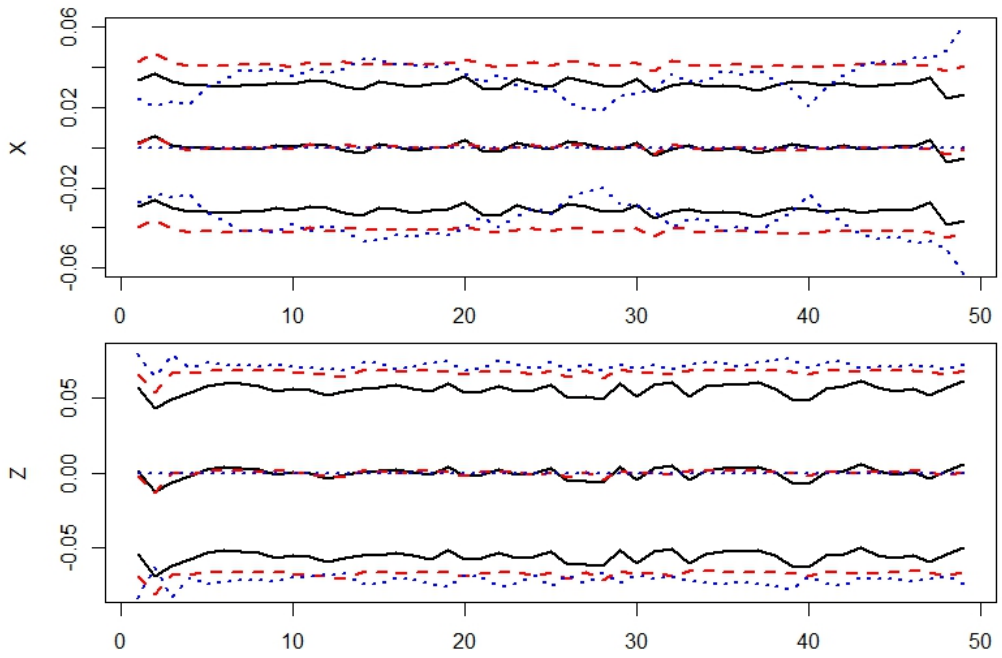

Figure 4.5: Confidence Intervals Comparison. Blue dotted line: sample quantile; Black solid line: GPRr estimation; Red dashed line: GPR1 estimation.

4.4.2 Heston model importance sampler

First of all, we should introduce some analytical results for Heston model derived by Lysy under the prior $\pi(\theta) \propto \gamma\sigma^2$. The posterior distribution of $\theta|Y$ is

$$\alpha|Y \sim \mathcal{N}(a, b),$$

$$\frac{c}{\tau^2}|\alpha, Y \sim \chi^2_{(f)}$$

$$(\gamma, \beta, \kappa)|\tau^2, \alpha, Y \sim \mathcal{N}_k(\mu, V)$$

where the coefficients can be found in Lysy (2012) [20].

Heston model's likelihood function can be written into the form like (4.2),

$$L(\theta|Y) \propto \prod_{i=1}^{N} p_{\Delta t}(Y_i|Y_{i-1}, \theta). \tag{4.19}$$

Since we introduce the missing data problem with k level of resolution. Let us simplify some definitions. We are using Δt as the time interval between the observations, then we define $\Delta t_k = \Delta t/2^k$. Thus we can relabel the observations as $Y = (Y_0, Y_1, \ldots, Y_n) = (Y_{k,0}, Y_{k,2^k}, \ldots, Y_{k,n2^k})$, with the missing data $Y_m = (Y_{k,1}, \ldots, Y_{k,2^k-1}, Y_{k,2^k+1} \ldots, Y_{k,n2^k-1})$. We assume the complete data to be $Y_k = Y \cup Y_m = (Y_{k,0}, \ldots, Y_{k,n2^k})$, so when $k \to \infty$, the true transition matrix can be approximated by this multi-Gaussian distribution:

$$Y_{k,n+1}|Y_{k,n}, \theta \approx N\left(\begin{pmatrix} X_{k,n} + (\alpha - \frac{1}{8}Z^2_{k,n})\Delta t_k \\ Z_{k,n} + (\beta/Z_{k,n} - \frac{1}{2}\gamma Z_{k,n})\Delta t_k \end{pmatrix} \begin{pmatrix} \frac{1}{4}Z^2_{k,n}\Delta t_k & \frac{1}{2}\rho\sigma Z_{k,n}\Delta t_k \\ \frac{1}{2}\rho\sigma Z_{k,n}\Delta t_k & \sigma^2\frac{1}{2}\rho\sigma Z_{k,n}\Delta t_k \end{pmatrix}\right) \tag{4.20}$$

then we will have an Euler likelihood $\hat{L}(\theta|Y_k)$ by approximate the true transition densities by Gaussian densities. So according to (4.10), we have approximately true density

$$\hat{p}(\theta, Y_m|Y) \propto \hat{L}(\theta|Y_k)\pi(\theta) \propto \prod_{i=1}^{n2^k} \hat{p}_{\Delta t}(Y_{k,i}|Y_{k,i-1}, \theta) \tag{4.21}$$

Now, let us focus on the importance sampler. Three importance samplers will be compared:

1. The first importance sampler is the oracle Gaussian distribution. Oracle method here means to construct the proposal Gaussian distribution using the mean and the

variance calculated using the true samples from MCMC method. Theoretically, it shows the best case scenario when we use Gaussian distribution as the proposal distribution. While we have to understand this sample data is usually not available, we get them using computationally expensive MCMC method, which we are trying to avoid using. The meaning of this oracle method is to test how good a Gaussian distribution can serve as the proposal distribution.

The importance sampler proposal for oracle method is the same as (4.13), but with $p_{oracle}(X_m|X)$ instead of $p_{GPR}(X_m|X)$.

2. The second importance sampler is directly from the result of multi-response GPR. As we discussed before, we fit multi-response GPR with $r = 1$ and get the joint estimation of the missing points. The importance sampler is described in (4.13). $p_{GPR}(X_m|X)$ is a 98 dimensional Gaussian density as we have 49 missing points for X_t and Z_t each.

3. The third importance sampler is based on GPR too. In last part, we found GPR estimation for X_t is not so good. Nonetheless, the estimations for Z_t are perfect. Actually we do not need GPR estimation for the X_t part because there is a close form conditional distribution for X_t given Z_t, every missing data x_i follows

$$x_i|Y, Z_m, \theta \sim \mathcal{N}(A_i, B_i),$$

where the coefficients can be found in Lysy (2012)[20].

In this GPR-Z method, the proposal importance sampling can be defined as

$$q(\theta, Y_m) = p(\theta|Y_m, Y)p(X_m|Z_m, Y)p(Z_m|Y) \tag{4.22}$$

where $Y_m = (X_m, Z_m)$, $p(X_m|Z_m, Y)$ is the missing X part's conditional density and $q(Z_m|Y)$ is the GPR estimation density. We still use multi-response GPR to fit the model and only keep the result of Z_m part, draw samples for Z_m and calculate the density value. Then we based on the sample of Z_m draw sample for Y_m and calculate the conditional density. In this way, we manage to only use the good estimated part of GPR, Z_t part.

The comparison procedure goes by firstly fit the GPR with $r = 1$ and get the joint estimation of the missing points. Then sample $20,000$ samples for every each of those three methods. For oracle method, we draw samples with MCMC data's mean and variance for both missing X_t and Z_t. For multi-response GPR method, we use multi-response GPR's

estimation results as the proposal density to sample from. For the GPR-Z method, we use multi-response GPR's Z_t estimation results to sample from Z_t part and then condition on Z_t to sample from X_t using (3). After we sample for the missing data, we based on those result to sample for θ, which is the same procedure for all three methods. At the same time, we calculate all the samples' density values. Using those values, we can analytically calculate the value for every proposal density $q(X_m, \theta)$.

A good proposal density should be easily sampled from and also the weights w_i, defined in section 4.2, should not be too different. We do not want to have one group of weights very large and the other groups quite small. In that case, we will keep getting samples from the high weight points, according to the importance sampler procedure. We want those weights to be reasonably distributed. That is what we are evaluating here using effective sample size (ESS) factor[2]:

$$ESS = \frac{1}{1 + var(w)/E^2(w)} \tag{4.23}$$

which is used here as a criterion here.

The scale of the numeric value of $p(\theta, Y_m|X)$ and $q(\theta, Y_m|X)$ can be quite small, especially when we have large observation number, which is 50 in our case. So it is wise to compute them in a log scale. In Figure 4.6, it shows the density for value $\log(p(\theta, Y_m|X)) - \log(q(\theta, Y_m|X))$ of all three methods. All the numeric values of target densities are all proportional to the true value. To make it easier to compare, all three sets of values are set to mean 0. The focus here should be drawn to the range of the density covers.

The dotted blue line represents the oracle method. It is the best Gaussian density and it yields a range of around 5, that means for almost all the log likelihood ratio, the largest one is around e^5 times larger than the smallest one. Actually it is quite good result, and we can get more than 10% on ESS from oracle method.

However, when we use the GPR method suggested in the previous part, we red solid curve and the result is no good. The density curve covers the range more than 15, which means that the larger r_i is e^{15} times larger than the smaller one. This is quite bad and the ESS value is far less than 1%. To find out the reason why the result is bad, we might blame the poor fit for X_t value as shown in Figure 4.5. Some samples on the edges can make a great different because as we can see the coverage of some GPR estimations are not great. Besides, GPR with r-exponential covariance function theoretically should not be performed on a non-stationary data because the r-exponential covariance's stationarity property[31].

[2]ESS factor lies between 0 to 1 and the close to 1, the better

Just for the computational simplification reason, we perform a multi-responses GPR on the data.

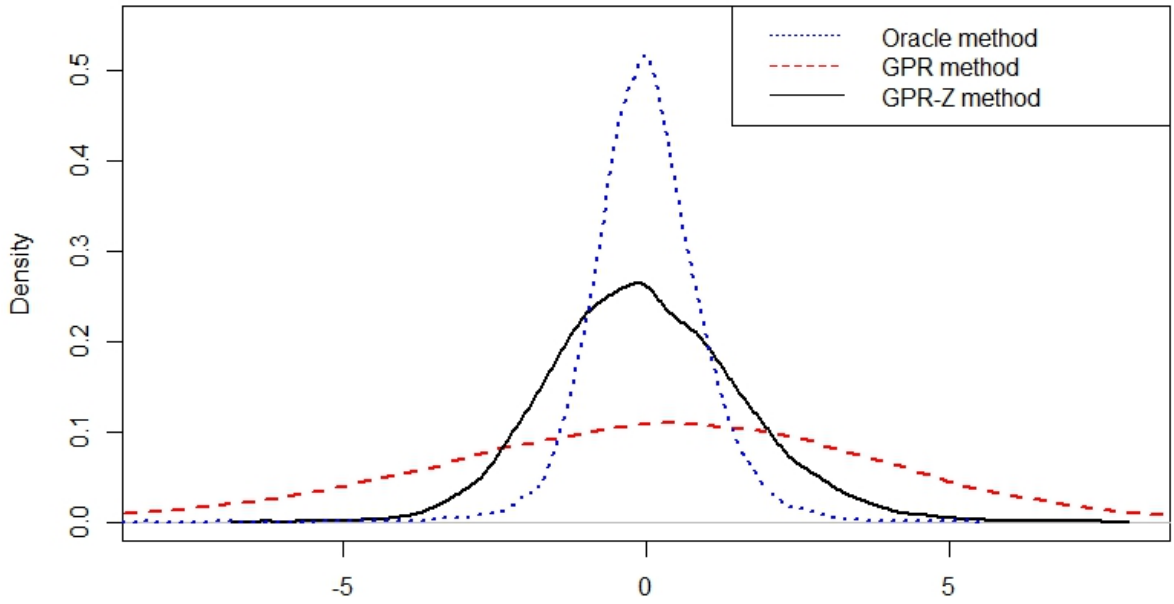

Figure 4.6: Log likelihood comparison with different approaches

The black dashed curve is for the GPR-Z approach. The density curve is much better than multi-response GPR. Although the density is not as concentrated as the oracle method, but the range is around 6 which is not too large. ESS confirms this result that yields value around 3%.

Method	Trial 1	Trial 2	Trial 3
Oracle	0.091471	0.179531	0.157764
GPR-Z	0.025035	0.021033	0.028071
GPR	0.000678	0.000219	0.001920

Table 4.1: ESS value for different approaches

The ESS value can be affected greatly by some outliers. So we run every method three times just to make sure the result is accurate. As we can see in Table 4.1 that the value can change a lot but the scales stay no change. Oracle is around 10%, GPR-Z method

is around 3% and multi-response GPR method is less than 0.1%. The major part of the log likelihood difference always focus on a range of 4 for oracle method, 7 for Half GPR method and 15 for total GPR method.

The result from the log likelihood ratio and ESS value suggest that we should try GPR-Z method. Although oracle method yields better result, but the data come at the cost of MCMC method, which we are trying to avoid. In order to compare with MCMC method, we use sample size of $500,000$ too.

The posterior estimations comparison is shown in the Figure 4.7. As we can see that, if we use k=1 situation as the true posterior, GPR-Z method achieves similar posterior densities. Compared with the simulate parameters values $\alpha = 1$, $\gamma = 5$, $\beta = 0.82$, $\sigma = 0.6$ and $\rho = -0.8$, the posterior estimations are reasonable. In addition, GPR-Z generates independent sample, rather highly correlated samples from MCMC as we shown before. The densities from GPR-Z are not smooth, because we sample only 20,000 samples using GPR-Z. So there are some parameters values cannot be covered or relatively sparsely sampled.

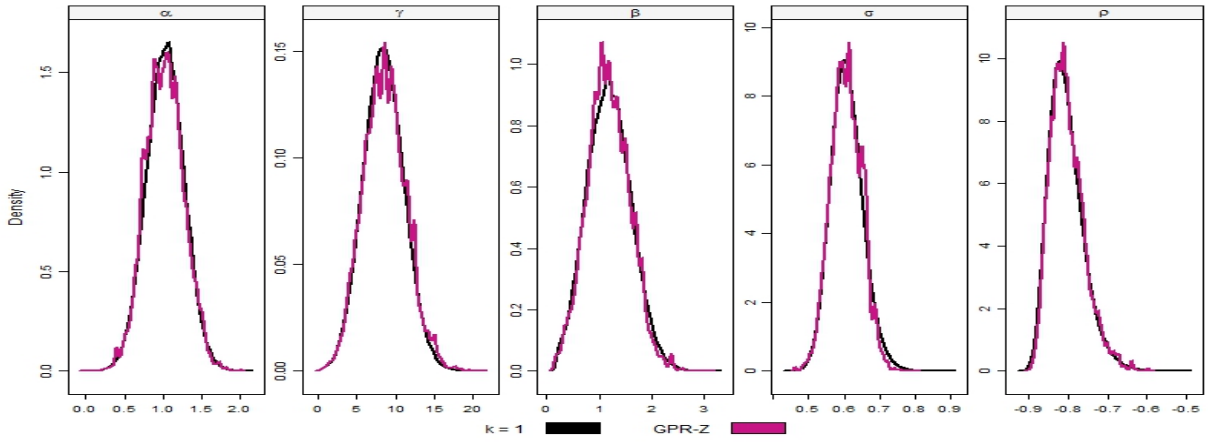

Figure 4.7: Parameters Posterior Comparison Between MCMC and GPR-Z

Although the ESS value is not great for GPR-Z method, but when we generate sample from GPR-Z method, we can get reasonably good posterior density. Noticed that random draw from Gaussian distribution is much faster than MCMC, GPR-Z method is reasonably faster.

Chapter 5

Conclusion

This thesis examined the possibility of using Gaussian process regression to perform inference for continuous time stochastic processes. Two applications were considered from which we draw the following conclusions:

1. Information about the characteristics of the true model can be very helpful to choose the right GPR covariance function. In chapter 3, the FHN model's example has shown us that we can achieve much better result by simply involving periodic term in the covariance function. All the shape turns in the true curve can be captured if we simply have some observations near it in any of the cycles. We even show that the future predictions of the FHN model can still capture the periodic property with considerably less error. In chapter 4, the Heston model example shows us that we should use the information, such as the Markov property, to modify our covariance. The results shows that GPR using r-exponential covariance function with $r = 1$, which has Markov property, gives better estimations than GPR using estimated r value. Also, in this example, we find out that we should be careful to apply stationary covariance function to non-stationary data like the price part of the Heston model. Here, we have exclusively focused on the r-exponential covariance function, but the potential advantages of other covariances remain to be explored.

2. When using multi-response GPR to model a multi-variate stochastic process, the dependence mostly appears in the conditional mean part rather than the conditional variance part. In chapter 3, the comparison between the multi-response GPR and simple GPR shows us that the conditional mean's curves from multi-response GPR are better than simple GPR, while the conditional variance is not. In chapter 4,

the Heston model shows us the good estimations for both mean and variance for Z_t, and good mean estimation but bad variance estimation for X_t. If we have the information about the model that one variable's variance is connected with another variable's mean, such as the Heston model, we can try to modify GPR methodology to estimate X_t based on Z_t.

3. GPR has great potential for the parameters inference for multivariate SDE's. MCMC approach based on Euler approximation and Gibbs sampling can give solutions, but it is computationally too expansive. Nevertheless, 3% ESS factor value for popular Heston model can be of tremendous practical significance. As we mentioned before, sample from Gaussian distribution is really fast. As we can see in the last part of chapter 4, GPR-Z method achieve reasonably better posterior estimation than MCMC method when same size of samples are used. In addition, there is still modification can be made to the methodology, such as the choice of the covariance function.

For the future work, there is still a lot to be done. Such as,

1. In practice, Heston model only has observations X_t, not Z_t. This proposed methodology would have to be adapted to impute Z_t based on X_t before the GPR approximation.

2. In the Heston model example, we find out that GPR estimation for X_t is not so good, which is because the variance of X_t depends on the value of Z_t. However, if we can modify GPR covariance function and include the information of Z_t for the variance of X_t, we might get an better estimation for X_t using GPR.

Overall, GPR with its advantages deserves more attentions in the study of inference for continuous stochastic process. With the possible adaptations, the potential of GPR importance sampler based parametric inference is still yet to be further explored.

APPENDICES

Appendix A

Profile Likelihood

In the one-response GPR, we the log-likelihood function is:

$$\ell(\theta) = -\frac{1}{2}Y'V^{-1}Y - \frac{1}{2}\log|V| \tag{A.1}$$

If we define

$$M = \left[\exp\left(\frac{-(x_i - x_j)^2}{2\lambda^2}\right)\right]_{n*n}, \quad i, j = 1, 2, \ldots, n \tag{A.2}$$

Then,

$$V^{-1} = \sigma_f^{-2}M^{-1} \tag{A.3}$$

and

$$|V| = n\sigma_f^2|M| \tag{A.4}$$

So we can simplified A.1 into:

$$\ell(\theta) = -\sigma_f^{-2}\frac{1}{2}Y'M^{-1}Y - \frac{1}{2}\log|M| - \frac{n}{2}\log\sigma_f^2 \tag{A.5}$$

To maximize the log-likelihood function over the scale of σ_f^2, which means:

$$\sigma_f^{-4}\frac{1}{2}Y'M^{-1}Y - \frac{n}{2\sigma_f^2} = 0 \Rightarrow \hat{\sigma}_f^2 = \frac{Y'M^{-1}Y}{n} \tag{A.6}$$

So if we are give the value for λ, we can get the profile estimation of σ_f^2 as shown above.

In multi-response GPR, when we are given the value for V as defined in section 2.4, we can make a transformation to Y and get $Z = U^{-1}Y$. So we will have the result

$$Z' = (Z'_1, \ldots, Z'_N) \Rightarrow Z_i \sim N(\mathbf{0}, \Sigma),\ i = 1, \ldots, N \tag{A.7}$$

where N is the number of the observation, because every row of Y, $Y_i \sim N(\mathbf{0}, V_{i,i}\Sigma)$.

So every Z_i is a realization of $N(\mathbf{0}, \Sigma)$. According to the MLE,

$$\log \prod_i f(Z_i|\Sigma) \propto \ell(\Sigma) = -\frac{N}{2}\log|\Sigma| - \frac{NM}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{N} Z_i \Sigma^{-1} Z'_i \tag{A.8}$$

and the matrix derivative rule,

$$\frac{\partial tr(AXB)}{\partial X} = BA, \qquad \frac{\partial \ln|aX|}{\partial X} = X^{-1} \tag{A.9}$$

where X is a matrix. Assume $\kappa = \Sigma^{-1}$ then we get

$$\frac{\partial l(\kappa)}{\partial \kappa} = \frac{N}{2}\kappa^{-1} - \frac{1}{2}\sum_{i=1}^{N} Z'_i Z_i = 0 \Rightarrow \hat{\kappa}^{-1} = \hat{\Sigma} = \frac{\sum_{i=1}^{N} Z'_i Z_i}{N} = \frac{Z'Z}{N} \tag{A.10}$$

so we will have the result that

$$\hat{\Sigma} = \frac{1}{N}Z'Z = \frac{1}{N}Y'V^{-1}Y = (\hat{\sigma}_{i,j})_{M\times M} \tag{A.11}$$

where $N\sigma_{i,j} = Y'_i V^{-1} Y_j$.

Noticed that the following properties of Kronecker Product can be useful of further simplify the likelihood function:

$$\det(A \otimes B) = [\det(A)]^b \times [\det(B)]^a$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

where A and B are random square matrix with the size of $a \times a$ and $b \times b$.

Now, let's take a look at the (2.18), and the first part of that can be simplify as

$$\begin{aligned}
\text{vec}(Y)'C^{-1}\text{vec}(Y) &= (Y_1, Y_2, \ldots, Y_M)'(\Sigma^{-1} \otimes V^{-1})(Y_1, Y_2, \ldots, Y_M) \\
&= (Y_1, Y_2, \ldots, Y_M)' \begin{pmatrix} \sigma'_{1,1}V^{-1} & \cdots & \sigma'_{1,M}V^{-1} \\ \vdots & \ddots & \vdots \\ \sigma'_{M,1}V^{-1} & \cdots & \sigma'_{M,M}V^{-1} \end{pmatrix} (Y_1, Y_2, \ldots, Y_M) \\
&= \sum_{i,j=1}^{M} \sigma'_{i,j} Y_i' V^{-1} Y_j = N \sum_{i,j=1}^{M} \sigma'_{i,j}\sigma_{i,j}
\end{aligned}$$

where $\sigma'_{i,j}$ is the (i,j) element of Σ^{-1} and $Y_i = (y_{1i}, y_{2i}, \ldots, y_{Ni})'$

The second part of (2.18) is

$$\begin{aligned}
\log|C| &= M\log|V| + N\log|\hat{\Sigma}| \\
&= M\log|V| + N\log\left|\frac{Y'V^{-1}Y}{N}\right|
\end{aligned}$$

Based on the profile MLE result in the previous part, we can simply (2.18) into

$$\ell(\theta) = -\frac{N}{2}\sum_{i,j=1}^{M}(\hat{\sigma}'_{i,j}\sigma_{i,j}) - \frac{1}{2}(M\log|V| + N\log\left|\frac{Y'V^{-1}Y}{N}\right|) \tag{A.12}$$

where $\hat{\sigma}'_{i,j}$ is the (i,j) element of $\hat{\Sigma}^{-1}$ and $Y_i = (y_{1i}, y_{2i}, \ldots, y_{Ni})'$.

Appendix B

Mathematical Deduction of Conditional Distribution

All the definition will be the same as we used in Chapter 2.

$$\Sigma^{-1} = \begin{bmatrix} K & K'_* \\ K_* & K_{**} \end{bmatrix}^{-1} = \begin{bmatrix} A & B' \\ B & C \end{bmatrix} \tag{B.1}$$

where K, K_* and K_{**} are the covariance matrix. K and A are $n \times n$ matrices; K_* and B are $1 \times n$ matrices and K_{**} and C are 1×1 matrices.

Based on the algebra knowledge about how to compute inverse matrix, we have

$$\begin{cases} A = (V - V'_* V_{**} V_*)^{-1} = V^{-1} + V^{-1} V'_* (V_{**} - V_* V^{-1} V'_*)^{-1} V_* V^{-1} \\ C = (V_{**} - V_* V V'_*)^{-1} = V_{**}^{-1} + V_{**}^{-1} V_* (V - V'_* V_{**}^{-1} V_*)^{-1} V'_* V_{**}^{-1} \\ B = -V_{**}^{-1} V_* (V - V'_* V_{**} V_*)^{-1} = -V^{-1} V'_* (V_{**} - V_* V V'_*)^{-1} \\ |\Sigma| = |V||V_{**} - V_* V^{-1} V'_*| \end{cases} \tag{B.2}$$

As we remember the joint probability density function for the n-dimension observation vector Y is

$$f_Y(Y) = \frac{1}{(2\pi)^{(n/2)}|K|} \exp\left[-\frac{Y' V^{-1} Y}{2}\right] \tag{B.3}$$

Now, if we have a point to estimate, we can view the observations and the expected

point as a sample from an $n+1$ dimension multivariate Gaussian. The joint probability density function for the observations and the prediction point should be

$$f_{(Y,y^*)}(Y,y^*) = \frac{1}{(2\pi)^{(n+1/2)}|\Sigma|}\exp\left[-\frac{(Y',y^*)'\Sigma^{-1}(Y',y^*)}{2}\right] \tag{B.4}$$

where (Y',y^*) is an $1\times n$ matrix, and Σ is defined in (B.1).

According to the definition of the conditional distribution and using the (B.1) and (B.2),

$$\begin{aligned} f_{y^*|Y}(y^*|Y) &= \frac{f_{(Y,y^*)}(Y,y^*)}{f_Y(Y)} \\ &= \frac{(2\pi)^{n/2}|V|}{(2\pi)^{(n+1)/2}|V||V_{**}-V_*V^{-1}V_*'|}\exp\left[-\frac{1}{2}[(Y',y^*)'\Sigma^{-1}(Y',y^*) - YK^{-1}Y']\right] \\ &= \frac{1}{(2\pi)^{1/2}|V_{**}-V_*V^{-1}V_*'|} * \exp(A) \end{aligned} \tag{B.5}$$

where $A = -\frac{1}{2}(y-V_*V^{-1}Y)(V_{**}-V_*V^{-1}V_*')^{-1}(y-V_*V^{-1}Y)$.

In conclusion,

$$y'|Y \sim N(K_*K^{-1}Y,\ K_{**}-K_*K_{-1}K_*') \tag{B.6}$$

References

[1] Sudipto Banerjee, Alan E Gelfand, Andrew O Finley, and Huiyan Sang. Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):825–848, 2008.

[2] Alexandros Beskos, Gareth Roberts, Andrew Stuart, and Jochen Voss. Mcmc methods for diffusion bridges. *Stochastics and Dynamics*, 8(03):319–350, 2008.

[3] Edwin Bonilla, Kian Ming Chai, and Christopher Williams. Multi-task gaussian process prediction. 2008.

[4] Phillip Boyle and Marcus Frean. Multiple output gaussian process regression. 2005.

[5] Andrew Butler, Thomas D Humphries, Pritam Ranjan, and Ronald D Haynes. Efficient optimization of the likelihood function in gaussian process modelling. *arXiv preprint arXiv:1309.6897*, 2013.

[6] John C Cox, Jonathan E Ingersoll Jr, and Stephen A Ross. A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society*, pages 385–407, 1985.

[7] Noel AC Cressie. Statistics for spatial data, revised edition, 1993.

[8] Pierre Delfiner et al. *GEOSTATISTICS: MODELING SPATIAL UNCERTAINTY*, volume 497. Wiley. com, 2009.

[9] Peter J Diggle and Paulo J Ribeiro. *MODEL-BASED GEOSTATISTICS*. Springer, 2007.

[10] Bjørn Eraker. Mcmc analysis of diffusion models with application to finance. *Journal of Business and Economic Statistics*, 19(2):177–191, 2001.

[11] William R Esposito and Christodoulos A Floudas. Deterministic global optimization in nonlinear optimal control problems. *Journal of Global Optimization*, 17(1-4):97–126, 2000.

[12] Danielle Florens-Zmirou. On estimating the diffusion coefficient from discrete observations. *Journal of applied probability*, pages 790–804, 1993.

[13] Mark Gibbs and David JC MacKay. Efficient implementation of gaussian processes. 1997.

[14] Robert B Gramacy and Herbert KH Lee. Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483), 2008.

[15] Mathieu Kessler, Alexander Lindner, and Michael Sorensen. *STATISTICAL METHODS FOR STOCHASTIC DIFFERENTIAL EQUATIONS*, volume 124. CRC Press, 2012.

[16] Juš Kocijan, Roderick Murray-Smith, Carl Edward Rasmussen, and Bojan Likar. *PREDICTIVE CONTROL WITH GAUSSIAN PROCESS MODELS*, volume 1. IEEE, 2003.

[17] SC Kou, Benjamin P Olding, Martin Lysy, and Jun S Liu. A multiresolution method for parameter estimation of diffusion processes. *Journal of the American Statistical Association*, 107(500):1558–1574, 2012.

[18] D.G Krige. A statistical approach to some mine valuations and allied problems at the witwatersrand, 1951.

[19] Malte Kuss and Carl E Rasmussen. Assessing approximations for gaussian process classification. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 699–706, 2005.

[20] Martin Lysy. *The Method of Batch Inference for Multivariate Diffusions*. PhD thesis, 2013.

[21] Kenneth Blake MacDonald. *GPfit: A New R Package for Fitting Gaussian Process Models to Deterministic Simulators*. PhD thesis, Acadia University, 2012.

[22] David JC MacKay. Gaussian processes-a replacement for supervised neural networks? 1997.

[23] Jay D Martin and Timothy W Simpson. Use of kriging models to approximate deterministic computer models. *AIAA journal*, 43(4):853–863, 2005.

[24] Georges Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

[25] R Murray-Smith and A Girard. Gaussian process priors with arma noise models. In *IRISH SIGNALS AND SYSTEMS CONFERENCE, MAYNOOTH*, pages 147–152, 2001.

[26] Radford M Neal. Priors for infinite networks. In *BAYESIAN LEARNING FOR NEURAL NETWORKS*, pages 29–53. Springer, 1996.

[27] Radford M Neal. Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026*, 1997.

[28] Luca Pasolli, Farid Melgani, and Enrico Blanzieri. Gaussian process regression for estimating chlorophyll concentration in subsurface waters from remote sensing data. *Geoscience and Remote Sensing Letters, IEEE*, 7(3):464–468, 2010.

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

[30] Pritam Ranjan, Ronald Haynes, and Richard Karsten. A computationally stable approach to gaussian process interpolation of deterministic computer simulation data. *Technometrics*, 53(4):366–378, 2011.

[31] Carl E Rasmussen and Christopher KI Williams. Gaussian processes for machine learning (adaptive computation and machine learning series), 2005.

[32] Carl Edward Rasmussen. Gaussian processes in machine learning. In *ADVANCED LECTURES ON MACHINE LEARNING*, pages 63–71. Springer, 2004.

[33] Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

[34] Carl Edward Rasmussen and Carl Edward Rasmussen. Evaluation of gaussian processes and other methods for non-linear regression. Technical report, 1996.

[35] Paulo J Ribeiro Jr and Peter J Diggle. geor: A package for geostatistical analysis. *R news*, 1(2):14–18, 2001.

[36] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989.

[37] Thomas J Santner, Brian J Williams, and William Notz. *THE DESIGN AND ANALYSIS OF COMPUTER EXPERIMENTS*. Springer, 2003.

[38] JQ Shi, B Wang, Roderick Murray-Smith, and DM Titterington. Gaussian process functional regression modeling for batch data. *Biometrics*, 63(3):714–723, 2007.

[39] Christopher KI Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In *LEARNING IN GRAPHICAL MODELS*, pages 599–621. Springer, 1998.

[40] Christopher KI Williams and David Barber. Bayesian classification with gaussian processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1342–1351, 1998.

[41] CKI Williams. Gaussian processes. 2002.

[42] Hugh R Wilson. *SPIKES, DECISIONS, AND ACTIONS: THE DYNAMICAL FOUNDATIONS OF NEUROSCIENCE*, volume 5. Oxford University Press Oxford, 1999.