Master Thesis
Computer Science
December 2013

Analytic Long Term Forecasting with Periodic Gaussian Processes

Author: Nooshin Haji Ghassemi

School of Computing
Blekinge Institute of Technology
37179 Karlskrona
Sweden

This thesis is submitted to the School of Computing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

Contact Information

Author: Nooshin Haji Ghassemi
E-mail: nooshin.hghs@gmail.com

External Advisor(s):
Dr. Marc Deisenroth
Department of Computing,
Imperial College London, United Kingdom
Prof. Jan Peters
Department of Computer Science,
Technische Universität Darmstadt, Germany

University advisor: Dr. Johan Holmgren
School of Computing and Communications

School of Computing
Blekinge Institute of Technology
371 79 KARLSKRONA SWEDEN

Internet: www.bth.se/com
Phone: +46 455 385000
SWEDEN

Abstract

In many application domains such as weather forecasting, robotics and machine learning we need to model, predict and analyze the evolution of periodic systems. For instance, time series applications that follow periodic patterns appear in climatology where the CO_2 emissions and temperature changes follow periodic or quasi-periodic patterns. Another example can be in robotics where the joint angle of a rotating robotic arm follows a periodic pattern. It is often very important to make long term prediction of the evolution of such systems.

For modeling and prediction purposes, Gaussian processes are powerful methods, which can be adjusted based on the properties of the problem at hand. Gaussian processes belong to the class of probabilistic kernel methods, where the kernels encode the characteristics of the problems into the models. In case of the systems with periodic evolution, taking the periodicity into account can simplifies the problem considerably. The Gaussian process models can account for the periodicity by using a periodic kernel.

Long term predictions need to deal with uncertain points, which can be expressed by a distribution rather than a deterministic point. Unlike the deterministic points, prediction at uncertain points is analytically intractable for the Gaussian processes. However, there are approximation methods that allow for dealing with uncertainty in an analytic closed form, such as moment matching. However, only some particular kernels allow for analytic moment matching. The standard periodic kernel does not allow for analytic moment matching when performing long term predictions.

This work presents an analytic approximation method for long term forecasting in periodic systems. We present a different parametrization of the standard periodic kernel, which allows us to approximate moment matching in an analytic closed form. We evaluate our approximate method on different periodic systems. The results indicate that the proposed method is valuable for the long term forecasting of periodic processes.

Keywords: Gaussian Process, Periodic Kernel, Long Term Forecasting.

Acknowledgments

I wish to express my enormous gratitude to my supervisors. First, Marc Deisenroth, from whom I learned a lot about Gaussian processes and many other things. I would like to thank him for his generosity in sharing his thoughts and knowledge with me.

I was very lucky that I could work on my thesis in Intelligent Autonomous Systems lab at Technische Universität Darmstadt. In this regard, I would like to thank Jan Peters for letting me be part of his great group as well as sharing with me his valuable experiences regarding the presentation skills.

Last but not least, I wish to thank Johan Holmgren at Blekinge Institute of Technology who accepted kindly to be my supervisor, apart from his insightful comments on many drafts of my thesis.

Contents

List of Figures

Symbols and Notation

Symbol	Meaning
a, b	scalars
$\boldsymbol{a}, \boldsymbol{b}$	vectors (bold lower case letters)
$\boldsymbol{A}, \boldsymbol{B}$	matrices (bold capital letters)
$[\boldsymbol{A}]$	square brackets denote matrices
a_i	the ith element of vector $\boldsymbol{a}$
A_{ij}	the jth element in the ith row of matrix $\boldsymbol{A}$
$\boldsymbol{I}$	the identity matrix
$\boldsymbol{y}^\top$	transpose of vector $\boldsymbol{y}$
$\lvert\boldsymbol{A}\rvert$	determinant of matrix $\boldsymbol{A}$
$y\lvert x$	conditional random variable
$\mathrm{diag}(\boldsymbol{a})$	a diagonal matrix with diagonal entries $\boldsymbol{a}$
$\mathrm{Tr}(\boldsymbol{A})$	trace of square matrix $\boldsymbol{A}$
$\boldsymbol{\mathcal{D}}$	matrix of training data
$\boldsymbol{X}$	matrix of input data
$\mathcal{GP}$	Gaussian process
$k(x, x'), \mathbb{C}(x, x')$	covariance function or kernel evaluated at x and x'
$\boldsymbol{K}$	covariance matrix $\boldsymbol{K}$
l	characteristic length-scale
$\boldsymbol{\theta}$	vector of hyperparameters (free parameters of kernel)
σ_ε^2	noise variance
$\mathbb{E}[\boldsymbol{x}]$	expectation of variable $\boldsymbol{x}$
$\mathbb{V}(\boldsymbol{x})$	variance of variable $\boldsymbol{x}$
$p(\boldsymbol{x})$	a probability density function
$\sim$	distributed according to; example: $x \sim \mathcal{N}(\mu, \sigma^2)$
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian (normal) distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$

Chapter 1

Introduction

There are many systems around us whose evolutions follow periodic patterns. Boosting and recession in economics, sleeping behavior of animals and walking or running of a humanoid robot are only a few examples among many others. The need to model, predict and analyze the evolution of such periodic systems appears in many scientific disciplines, such as signal processing, control, and Machine learning. Ideally, one needs to build representative models that allow for precise prediction of the evolution of such systems. Sometimes we even need to make predictions long time ahead, e.g. to make informed decisions in advance.

1.1 Background and Related Work

For modeling purposes, *Gaussian processes* (GPs) are the state-of-the-art methods in the machine learning community [1, 2]. We are interested in this class of models for two main reasons. Firstly, GPs do not merely make pre-

Figure 1.1: Animals exhibit many periodic tasks, such as winging, walking or running.

dictions but also can express the uncertainty associated with the predictions. This is especially important when predicting ahead in time (See Chapter 3). GPs are also flexible models that can explicitly encode high-level prior assumptions regarding the systems into the models. Often assumptions such as smoothness and stationarity are made, see Section 2.1.1. The ingredient of GPs that allows encoding the assumptions into the models, is the kernel. The parametric form of the kernels imposes different characteristics on the models. In case of a periodic system, a *periodic kernel* allows building powerful models [3]. For instance, periodic kernels are used by Durrande et al. [4] to detect periodically expressed genes and by Reece and Roberts [5] in the context of target tracking. Rasmussen and Williams [2] use the periodic kernel to capture the periodic pattern of the CO_2 accumulation in the atmosphere. This thesis is particularly concerned with the use of *periodic Gaussian processes* for long term forecasting of the periodic systems.

MacKay [6] has proposed a periodic kernel, which is capable of capturing the periodicity of the patterns. In this work, we refer to it as the *standard* periodic kernel. Non-linear kernels such as the standard periodic kernel require approximations when it comes to long term forecasting with GPs, see Chapter 3. The approximation in general can be based on either numerical methods, see e.g. [7], or on analytic closed-form computations. Numerical solutions are easy to implement but they can be computationally demanding. An analytic solution based on moment matching (see Chapter 3) has been proposed by Quinonero-Candela et al.[8] for long term forecasting with GPs with the Gaussian kernel. Gaussian and polynomial kernels allow an analytic approximation for long term forecasting [8]. However, analytic moment matching is intractable for the standard periodic kernel.

1.2 Contribution

Our contribution is to propose a *double approximation* method, which provides an analytic solution for long term forecasting with periodic GPs. The key idea is to re-parametrize the standard periodic kernel in a way that allows analytic approximate inference. For re-parametrization we exploit the fact that analytic moment matching is possible for the Gaussian kernels. Furthermore, we evaluate our double approximation method empirically.

In particular, we aim to answer the following research questions:

1. How robust is the proposed double approximation against varying the test input distribution in one-step predictions?

2. How well does the Gaussian approximation with the proposed periodic

kernel (double approximation) perform in comparison to the same approximation method with the Gaussian kernel, when applied to long-term forecasting of a periodic system?

3. How does the double approximation method perform when applied to the long term prediction of periodic systems?

1.3 Outline

Chapter 2 presents the necessary background on Gaussian processes, with emphasis on the prediction. Furthermore, this chapter introduces the Gaussian and the standard periodic kernels as well as their roles in GP modeling. Different properties of the kernels and their performance on prediction of periodic systems is discussed.

Chapter 3 presents the main contribution of the thesis. First, we discuss the concepts of the long term forecasting. Then, our proposed double approximation method for long term forecasting of periodic systems is discussed.

In **Chapter 4**, the methods of the previous chapters are applied to the prediction of periodic systems. We empirically evaluate our double approximation method for the one-step prediction as well as the long term prediction of periodic systems. The results indicate that the proposed periodic kernel surpasses the non-periodic ones in prediction of the periodic systems.

Chapter 2

Introduction to Gaussian Processes

This chapter provides an overview of Gaussian process regression. In the first section, we present how to utilize GPs for regression and how to predict unknown continuous function values. In section 2.1.1, we introduce some commonly used covariance functions and discuss their properties. In the last section, we discuss model learning in the GPs.

2.1 Gaussian Process Regression

Regression is the problem of estimating real valued function values from inputs [9, 2]. In this section, we review the Bayesian treatment of the regression problem. For the underlying function f, the regression model becomes

$$y = f(\boldsymbol{x}) + \varepsilon, \quad \boldsymbol{x} \in \mathbb{R}^D, y \in \mathbb{R} \tag{2.1}$$

where $\boldsymbol{x}$ is the input vector, y is the observed target value and $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is additive identically independently distributed (i.i.d.) Gaussian noise with variance σ_ε^2. In Figure 2.1, the red crosses denote the noisy observed data points, called training data. The blue line in Figure 2.1 shows the underlying function. Note that the blue line is actually a finite number of data points, which we display as a line. The shaded area denotes the uncertainty associated with prediction at data points $\boldsymbol{x}$. From the figure, it is clear that the uncertainty shrinks near the observed data points. This happens because the observed data points provide information about the true function values for the regression model.

Suppose $\boldsymbol{f} = (f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots)$ is a vector of random variables. In this sense, $\boldsymbol{f}$ is a Gaussian process if the joint distribution over any finite subset

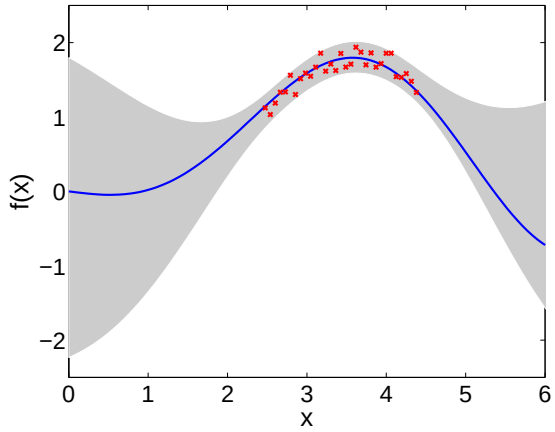


Figure 2.1: Example of a regression problem. The horizontal axis represents the inputs to the function f. The vertical axis represents the function values evaluated at the input points. Observed data points are marked by red crosses. The blue line illustrates the underlying function f. The shaded area represents the uncertainty associated with the estimation of the function values.

of the random variables $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)$ becomes a multivariate Gaussian distribution

$$p(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2). \tag{2.2}$$

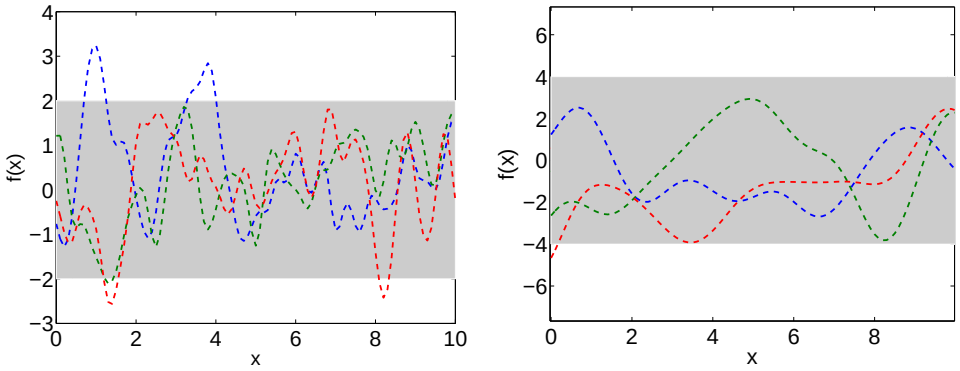
We can look at the Gaussian process as a generalization of a Gaussian distribution. While a Gaussian distribution can be fully characterized by its mean and variance, a Gaussian process can be attributed by the mean function $\mathbb{E}[f(\boldsymbol{x})] = m(\boldsymbol{x})$ and the covariance function $\mathbb{C}(f(\boldsymbol{x}), f(\boldsymbol{x}')) = k(\boldsymbol{x}, \boldsymbol{x}')$ such that

$$f \sim \mathcal{GP}(m, k). \tag{2.3}$$

2.1.1 Covariance Functions

Covariance functions or kernels play an important role in the GP modeling. A covariance function gives the correlation between function values, corresponding to the inputs $k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{C}(f(\boldsymbol{x}), f(\boldsymbol{x}'))$. Kernels have different parametric forms, which impose particular assumptions upon the functions, e.g. smoothness or stationarity assumptions. In the following, we introduce some commonly used kernels and discuss their properties.

The Gaussian kernel (Squared Exponential kernel) may be the most widely-used kernel in the Machine Learning community [2] due to its properties such as smoothness and stationarity. A smooth function suggests that



(a) $\alpha^2 = 1, \Lambda = 0.25$ (b) $\alpha^2 = 4, \Lambda = 1$

Figure 2.2: Three sample functions drawn at random from the prior distribution with the Gaussian kernel with different hyper-parameters. The higher length-scale (b) leads to smoother functions. Also note the difference of the vertical height of functions caused by different signal variance α^2 hyper-parameters.

if two data points are close in the input space, then the corresponding function values are highly correlated. A stationary kernel is a function of $\boldsymbol{x} - \boldsymbol{x}'$. A Gaussian kernel is defined as

$$k_{SE}(\boldsymbol{x}, \boldsymbol{x'}) = \alpha^2 \exp\big(-\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{x'})^\top \boldsymbol{\Lambda}^{-1}(\boldsymbol{x} - \boldsymbol{x'})\big), \quad \boldsymbol{x}, \boldsymbol{x'} \in \mathbb{R}^D, \tag{2.4}$$

where $\boldsymbol{\Lambda} = \text{diag}[l_1^2, \ldots, l_D^2]$ and α^2 denotes the signal variance that controls the vertical scale of the variation of the function. We call the parameters of the covariance function hyper-parameters. l_i are called length-scale hyper-parameters and control the degree of smoothness of the function. Figure 2.2 illustrates two GPs with Gaussian kernels with different hyper-parameter sets. The comparison of Figure 2.2a and 2.2b shows that larger length-scales lead to a smoother function.

It is clear from eq. (2.4) that the Gaussian kernel is stationary. It means that the covariance between two function values does not depend on the values of the corresponding input points, but only on the distance between them.

Although stationarity may be a desired property for many applications, it is restrictive in some cases. Figure 2.3 shows a periodic function, which a GP with Gaussian kernel fails to model appropriately. A function is periodic if it repeats on intervals called periods. The repeated parts have strong correlation with each other, regardless of their distance. For example, in Figure 2.3, the points on top of the waves have the same function values all

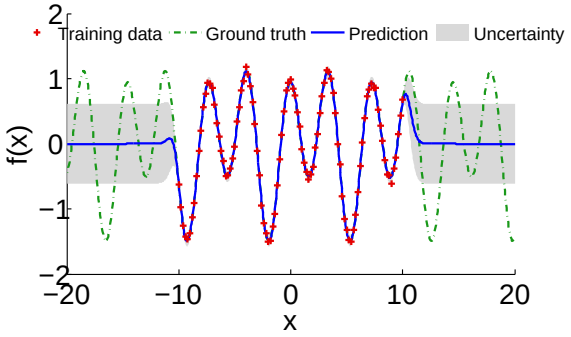


Figure 2.3: Prediction of a sin signal with the Gaussian kernel. While the red crosses represent the training set, the blue line represents the GP model prediction. The shaded area shows the 95% confidence intervals. Note that the shaded area enlarges at test points far from the training points, proving that the Gaussian kernel fails to extrapolate from the training set to the test set.

over the function, due to the periodicity. Stationarity in general cannot capture such a relation, in that it deduces the correlation between data points only from their distances. Furthermore, in Figure 2.3 the shaded area grows drastically for test points far from the training points, which indicates that the Gaussian kernel fails to extrapolate from the training set to the test set. Such periodic problems demand for more powerful kernels, which can handle the periodicity property.

MacKay [6] proposed a periodic kernel, which can capture the periodicity of the signals

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \alpha^2 \exp\left(\frac{-2\ \sin^2\left(\frac{a\boldsymbol{x}-a\boldsymbol{x'}}{2}\right)}{l^2}\right), \tag{2.5}$$

where l and α^2 have the same role as they have in case of the Gaussian kernel. The additional parameter a is related to the periodicity. Figure 2.4 illustrates the performance of the periodic kernel for modeling the simple periodic signal. The comparison of figure 2.3 and 2.4 reveals the advantages of the periodic kernel over non-periodic one on modeling periodic functions. Figure 2.4 illustrates that the GP with periodic kernel can predict test points correctly with very small uncertainty.

2.1.2 Prior Distribution

In the Bayesian setting, a Gaussian process can be seen as a probability distribution over functions $p(f)$. Figure 2.5a shows three sample functions

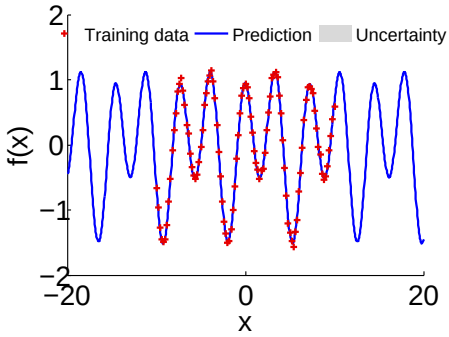
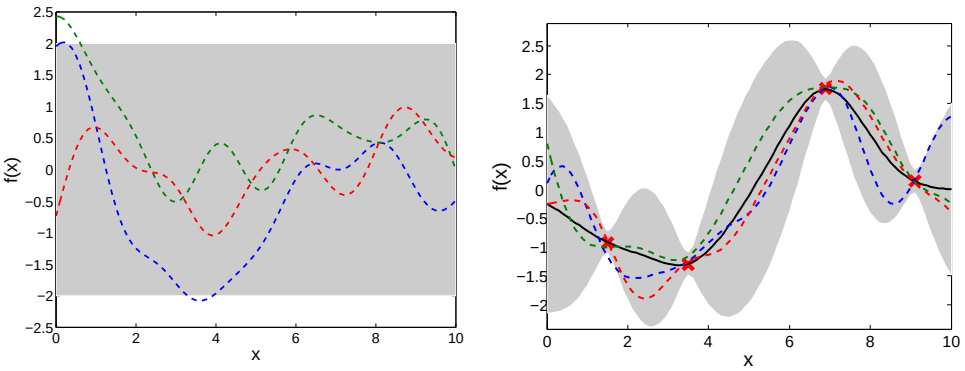


Figure 2.4: Prediction of a periodic signal with the periodic Gaussian process. The GP with periodic kernel can successfully extrapolate from the training set to the test set. The shaded area is almost zero throughout the model.

drawn at random from the prior distribution $p(f)$ specified by a particular GP. The prior probability tells us about the form of the functions which are more likely to represent the underlying function, before we observe any data points [2]. In Figure 2.5a, we assume a zero mean prior distribution. It means that if we keep drawing random functions from the distribution, the average of the function values becomes zero for any x. In Figure 2.5a, the shaded area is constant all over the function space, which demonstrates that the prior variance does not depend on x.

2.1.3 Posterior Distribution

We are not primarily interested in random functions drawn from the prior distribution, but the functions that represent our observed data points. In the Bayesian framework, it means moving from the prior distribution to the posterior distribution. Our observed data points combined with the prior distribution lead to the posterior distribution. Figure 2.5b illustrates what happens if we know the function values at some particular points. The figure illustrates wherever there is no observation the uncertainty increases. If more observed points are available, the mean function tends to adjust itself to pass through the observed points and the uncertainty reduces close to these points. Note that since our observations are noisy, the uncertainty is not exactly zero at the observed data.



(a) Prior (b) Posterior

Figure 2.5: Panel (a) shows three sample functions drawn at random from the prior distribution. The shaded area represents the prior variance and is constant all over the function space. Panel (b) shows three sample functions drawn randomly from the posterior distribution. The shaded region denotes twice the standard deviation at each input value x. The observed points are marked by red dots. Uncertainty shrinks near the observed data and increases for data points far from the observations.

2.2 Evidence Maximization

We can exploit the training data to directly learn the free parameters $\boldsymbol{\theta}$ (parameters of the covariance function) of the model. The log marginal likelihood (or evidence) (see e.g. [2]) is given by[1]

$$\mathcal{L}(\boldsymbol{\theta}) = \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = -\tfrac{1}{2}\boldsymbol{y}^T\boldsymbol{C}^{-1}\boldsymbol{y} - \tfrac{1}{2}\log|\boldsymbol{C}| - \tfrac{D}{2}\log(2\pi), \tag{2.6}$$

where $\boldsymbol{C} = \boldsymbol{K}_{\boldsymbol{\theta}} + \sigma^2\boldsymbol{I}$ and $|\boldsymbol{C}|$ is the determinant of the matrix $\boldsymbol{C}$. $\boldsymbol{K}_{\boldsymbol{\theta}}$ refers to the covariance matrix which depends on the values of hyper-parameters $\boldsymbol{\theta}$. Matrix $\boldsymbol{X}$ and vector $\boldsymbol{y}$ denote the training inputs and noisy observations, respectively. In the last term, D denotes the data dimension.

The goal is to find a set of free parameters $\boldsymbol{\theta}$ that maximizes the log marginal likelihood (evidence maximization). For evidence maximization, we compute the derivatives[2] of $\mathcal{L}(\boldsymbol{\theta})$ with respect to each hyper-parameter

$$\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\theta_j} = \tfrac{1}{2}\boldsymbol{y}^T\boldsymbol{C}^{-1}\frac{\partial\boldsymbol{C}}{\partial\theta_j}\boldsymbol{C}^{-1}\boldsymbol{y} - \tfrac{1}{2}\mathrm{Tr}\left[\boldsymbol{C}^{-1}\frac{\partial\boldsymbol{C}}{\partial\theta_j}\right]. \tag{2.7}$$

[1] We usually work with the log of the marginal likelihood. The marginal likelihood is the product of many small probability values. This can easily cause computational problems. By taking the log, the product transforms to the sum of the log of probabilities.

[2] For more on the matrix derivatives refer to Appendix A.1.3.

From eq. (2.7), it is clear that the computation of the derivatives depends on the parametric form of the covariance function k.

2.3 Prediction at a Test Input

Most commonly, a model is used for making prediction at new points with unknown targets. Suppose we have a training points set $\boldsymbol{\mathcal{D}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$, consisting of n training input and output pairs. There is also a test input $\boldsymbol{x}_*$[3] with unknown target. From the definition of the Gaussian process, the joint distribution of the function values at the training and test points is normally distributed, see Appendix A.1,

$$p(f_*, \boldsymbol{y}|\boldsymbol{x}_*, \boldsymbol{X}) = \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K} + \sigma_\varepsilon^2 \boldsymbol{I} & k(\boldsymbol{X}, \boldsymbol{x}_*) \\ k(\boldsymbol{x}_*, \boldsymbol{X}) & k(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{bmatrix}\right), \tag{2.8}$$

where $\boldsymbol{K}$ denotes the covariance matrix between each pair of observed points. Here, we assume the prior on f_* has a zero mean.

We can make predictions by conditioning the joint distribution $p(f_*, \boldsymbol{y})$ on the observation set $\boldsymbol{\mathcal{D}} = \{(\boldsymbol{X}, \boldsymbol{y})\}$, see Appendix A.1. In a GP, the predictive distribution is Gaussian with mean and covariance

$$\mu(\boldsymbol{x}_*) = k(\boldsymbol{x}_*, \boldsymbol{X})[\boldsymbol{K} + \sigma^2 \boldsymbol{I}]^{-1} \boldsymbol{y}, \tag{2.9}$$

$$\sigma^2(\boldsymbol{x}_*) = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - k(\boldsymbol{x}_*, \boldsymbol{X})[\boldsymbol{K} + \sigma^2 \boldsymbol{I}]^{-1} k(\boldsymbol{X}, \boldsymbol{x}_*), \tag{2.10}$$

respectively.

2.3.1 Multivariate Prediction

So far, we discussed about one dimensional targets $y \in \mathbb{R}$. If the targets has multiple dimensions $\boldsymbol{y} \in \mathbb{R}^E$, we train an independent GP for each target dimension. In other words, we train E models based on the same training data $\boldsymbol{X}$ but different output targets $\{y_i\}_{i=1}^E$. In such a case, we assume that the models are conditionally independent given the data set $\boldsymbol{X}$ [10]. Hence, the mean and the variance of the function values for each dimension are computed separately based on eq. (2.9) and eq. (2.10). For a given multivariate test input $\boldsymbol{x}_*$, the predictive distribution is a multivariate Gaussian with mean and covariance

$$\boldsymbol{\mu}_* = [m_{f_1}(\boldsymbol{x}_*) \ \ldots \ m_{f_E}(\boldsymbol{x}_*)]^\top, \tag{2.11}$$

$$\boldsymbol{\Sigma}_* = \mathrm{diag}([\sigma_{f_1}^2 \ \ldots \ \sigma_{f_E}^2]), \tag{2.12}$$

respectively.

[3] Test points are shown with a subscript asterisk.

Chapter 3

Prediction at Uncertain Inputs

In the previous chapter, we discussed how to predict at a deterministic input with a GP model. In this chapter, we investigate what happens if our observations are subject to uncertainty. Uncertainty may arise due to different reasons. For instance, a long term prediction of state evolution of a system $p(\boldsymbol{x}_1)$, $p(\boldsymbol{x}_2)$,... needs to effectively deal with uncertainty, since the inputs to the GPs are not deterministic points but uncertain data points.

In long term forecasting, we need to predict ahead in time, up to a specific time horizon. One way to achieve this is to iteratively compute one-step ahead prediction. In each step, the predictive distribution is given by $p(\boldsymbol{x}_{t+l+1}|\boldsymbol{x}_{t+l})$, where $\boldsymbol{x}_{t+l}$ serves as the input $\boldsymbol{x}_*$ and $\boldsymbol{x}_{t+l+1}$ plays the role of the target $f(\boldsymbol{x}_*)$. In such a setting, the input $p(\boldsymbol{x}_*)$ to the GP model is a probability distribution, not a deterministic point. As a result, the predictive distribution is obtained by

$$p(f(\boldsymbol{x}_*)) = \iint p(f(\boldsymbol{x}_*)|\boldsymbol{x}_*)p(\boldsymbol{x}_*)\mathrm{d}f\mathrm{d}\boldsymbol{x}_*, \tag{3.1}$$

which requires to ingrate over test input $\boldsymbol{x}_*$. Since $p(f(\boldsymbol{x}_*))$ is a complicated function of $\boldsymbol{x}_*$, the integral is analytically intractable. In general, the predictive distribution $p(f(\boldsymbol{x}_*))$ is not a Gaussian. However, if the input $p(\boldsymbol{x}_*)$ is Gaussian distributed, then the predictive distribution can be approximated by a Gaussian by means of moment matching.

Moment matching consists of computing only the predictive mean and covariance of $p(f(\boldsymbol{x}_*))$, i.e., the first two moments of the predictive distribution. Figure 3.1 illustrates moment matching with GPs when the input is normally distributed. The shaded area in the left panel denote the exact predictive distribution, which is not Gaussian and unimodal. The blue line,

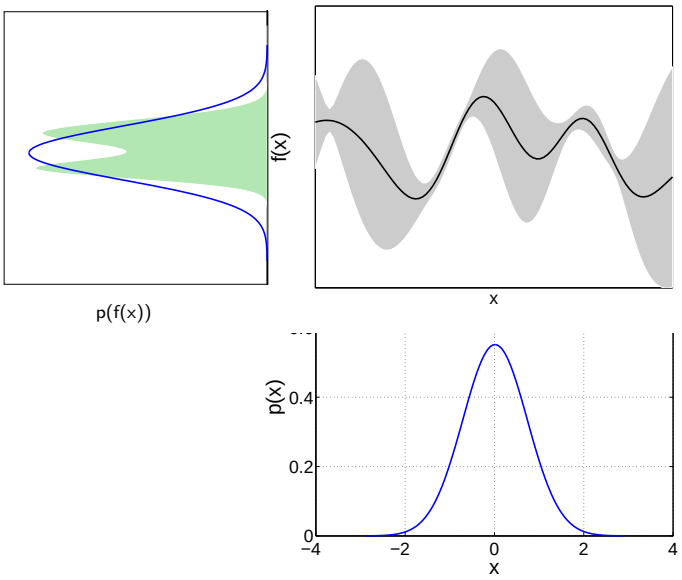


Figure 3.1: The bottom panel shows the Gaussian input, which is mapped through the GP model (upper-right panel). The shaded area (left panel) is the exact non-Gaussian output and the blue line is the result of the Gaussian approximation.

in the left panel, represents the predictive distribution computed by moment matching [11].

The exact Gaussian approximation is not analytically tractable for all forms of the kernels. Gaussian and polynomial kernels are among kernels that make the exact approximation possible [11]. On the contrary, moment matching with the standard periodic kernel in eq. (2.5) is analytically intractable. In this thesis, we present another parametric form of the standard periodic kernel, which in combination of a double approximation, allows for analytic long term forecasting of evolution of periodic systems.

3.1 Moment Matching with Gaussian Processes

Here, we detail the computation of the first two moments of the predictive distribution with Gaussian processes, that is largely based on the work by Deisenroth et al. [12]. Assume that input is normally distributed $p(\boldsymbol{x}_*) = \mathcal{N}(\boldsymbol{x}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$. From the law of iterated expectations (see Appendix A.2)

the mean of $p(f(\boldsymbol{x}_*))$, in eq. (3.1) becomes

$$m(\boldsymbol{x}_*) = \mathbb{E}_{\boldsymbol{x}_*}[\mathbb{E}_f[f(\boldsymbol{x}_*)|\boldsymbol{x}_*]] = \mathbb{E}_{\boldsymbol{x}_*}[\mu(\boldsymbol{x}_*)], \tag{3.2}$$

where $\mu(\boldsymbol{x}_*)$ is the mean function of the GP evaluated at $\boldsymbol{x}_*$. By plugging in eq. (2.9) for the predicted mean, we obtain

$$m(\boldsymbol{x}_*) = \boldsymbol{\beta}^\top \int k(\boldsymbol{X}, \boldsymbol{x}_*) p(\boldsymbol{x}_*) \mathrm{d}\boldsymbol{x}_*, \tag{3.3}$$

where $\boldsymbol{\beta} = (\boldsymbol{K} + \sigma_\varepsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y}$ and $\boldsymbol{X}$ is the matrix of training inputs.

In the same manner, the predictive variance can be obtained by

$$\begin{aligned} v(\boldsymbol{x}_*) &= \mathbb{E}_{\boldsymbol{x}_*}[\mathbb{V}_f[\boldsymbol{f}_*|\boldsymbol{x}_*]] + \mathbb{V}_{\boldsymbol{x}_*}[\mathbb{E}_{\boldsymbol{f}}[\boldsymbol{f}_*|\boldsymbol{x}_*]] \\ &= \mathbb{E}_{\boldsymbol{x}_*}[\sigma^2(\boldsymbol{x}_*)] + \mathbb{E}_{\boldsymbol{x}_*}[\mu(\boldsymbol{x}_*)^2] - m(\boldsymbol{x}_*)^2, \end{aligned} \tag{3.4}$$

where $\mu(\boldsymbol{x}_*)$ is given in eq. (2.9). The last term contains the predictive mean given in eq. (3.3). Plugging in the GP mean and variance from equations (2.9) and (2.10), the first two terms in eq. (3.4) are given as

$$\begin{aligned} \mathbb{E}_{\boldsymbol{x}_*}[\mu(\boldsymbol{x}_*)^2] &= \int \mu(\boldsymbol{x}_*)^2 p(\boldsymbol{x}_*) d\boldsymbol{x}_* \\ &= \boldsymbol{\beta}^\top \int k(\boldsymbol{X}, \boldsymbol{x}_*) k(\boldsymbol{x}_*, \boldsymbol{X}) p(\boldsymbol{x}_*) \mathrm{d}\boldsymbol{x}_* \boldsymbol{\beta}, \end{aligned} \tag{3.5}$$

and

$$\begin{aligned} \mathbb{E}_{\boldsymbol{x}_*}[\sigma^2(\boldsymbol{x}_*)] = &\int k(\boldsymbol{x}_*, \boldsymbol{x}_*) p(\boldsymbol{x}_*) d\boldsymbol{x}_* \\ &- \int k(\boldsymbol{x}_*, \boldsymbol{X})(\boldsymbol{K} + \sigma_\varepsilon^2 \boldsymbol{I})^{-1} k(\boldsymbol{X}, \boldsymbol{x}_*) p(\boldsymbol{x}_*) \mathrm{d}\boldsymbol{x}_*. \end{aligned} \tag{3.6}$$

The integrals in equations (3.4), (3.5), and (3.6) depend on the parametric form of the kernel function k. There is no analytic solution for these integrals with the standard periodic kernel in eq. (2.5). In the next section, we present a re-parametrization of the standard periodic kernel, which allows for an analytic approximation of these integrals. In particular, we propose a double approximation method to analytically compute the predictive mean and variance at uncertain points, by exploiting the fact that the involved integrals can be solved analytically for the Gaussian kernel.

3.1.1 Re-parametrization of Periodic Kernel

For notational convenience, we consider one dimensional inputs x in the following. Our periodic kernel uses a nonlinear transformation $u = (\sin(x),$

$\cos(x)$) of the inputs x and is given by[1]

$$k_{per}(x, x') = k_{\text{SE}}(u(x), u(x')) = \alpha^2 \exp\left(-\tfrac{1}{2}\boldsymbol{z}^\top \boldsymbol{\Lambda}^{-1}\boldsymbol{z}\right), \tag{3.7}$$

where

$$\boldsymbol{z} = \begin{bmatrix} \sin(ax) - \sin(ax') \\ \cos(ax) - \cos(ax') \end{bmatrix},$$

and $\boldsymbol{\Lambda} = \text{diag}[l_1^2, l_2^2]$, where we assume that $l = l_1 = l_2$, such that the sin and cos terms are scaled by the same value. The length scales l_i and signal variance α^2 play the same role as in the Gaussian kernel. The hyper-parameter a denotes periodicity, which in the case of the one dimensional input it is a scalar.

The periodic kernel in eq. (3.7) is just another representation of the standard periodic kernel in eq. (2.5). To prove this claim, let us ignore the diagonal scaling matrix $\boldsymbol{\Lambda}$ in eq. (3.7) for a moment. Multiplying out $\frac{1}{2}\boldsymbol{z}^\top\boldsymbol{z}$ yields

$$\tfrac{1}{2}\boldsymbol{z}^\top\boldsymbol{z} = 1 - \sin(ax)\sin(ax') - \cos(ax)\cos(ax'). \tag{3.8}$$

With the identity

$$\cos(x - x') = \cos(x)\cos(x') + \sin(x)\sin(x')$$

we obtain $\frac{1}{2}\boldsymbol{z}^\top\boldsymbol{z} = 1 - \cos(a(x - x'))$. Now, we apply the identity $\cos(2x) = 1 - 2\sin^2(x)$ and obtain

$$\tfrac{1}{2}\boldsymbol{z}^\top\boldsymbol{z} = 2\sin^2\left(\tfrac{a(x-x')}{2}\right).$$

Incorporating the scaling l from eq. (3.7) yields

$$\exp\left(-\tfrac{1}{2}\boldsymbol{z}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{z}\right) = \exp\left(-\frac{2\sin^2\left(\frac{a(x-x')}{2}\right)}{l^2}\right).$$

We see that our proposed kernel in eq. (3.7) is equivalent to the standard periodic kernel in eq. (2.5).

The extension to the multivariate input $\boldsymbol{x} \in \mathbb{R}^D$ is straightforward. We consider different length-scales for different input dimensions $\{l_i\}_{i=1}^D$. In the case of the multivariate inputs, the periodic kernel in eq. (3.7) becomes

$$k_{per}(\boldsymbol{x}, \boldsymbol{x}') = k_{\text{SE}}(u(\boldsymbol{x}), u(\boldsymbol{x}')) = \alpha^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D} \boldsymbol{z}_d^\top \boldsymbol{\Lambda}_d^{-1}\boldsymbol{z}_d\right), \tag{3.9}$$

where $\boldsymbol{z}_d$ is the d^{th} dimension of the trigonometrically transformed input and $\boldsymbol{\Lambda}_d = \text{diag}[l_d^2, l_d^2]$. Following the approach used for one dimensional case, we can obtain the standard periodic kernel for the multivariate input.

[1]Whenever it is necessary to distinguish the periodic kernel from the Gaussian kernel, they are denoted by k_{per} and k_{SE}, respectively.

3.1.2 Approximate Inference with a Periodic Kernel

Here we present our proposed approximate inference method for long term forecasting, which utilizes the periodic kernel in eq. (3.7). In particular, this parametrization of the standard periodic kernel allows for analytic approximation of the intractable integrals in equations (3.2), (3.5), and (3.6). Figure 3.2 illustrates our proposed approximation method. The goal is to compute the desired predictive distribution from a Gaussian distributed input. The top red line shows that there is no analytic solution for the Gaussian approximation by the standard periodic kernel. Instead, we propose the double approximation at the bottom of the figure, which contains two analytic approximations. In the first step, the first two moments of the input $p(\boldsymbol{x})$ are mapped to the trigonometric space $p(u(\boldsymbol{x}))$. Subsequently, the transformed input $p(u(\boldsymbol{x}))$ is mapped through the GP with a Gaussian kernel. In the following, we discuss both steps in detail.

3.1.3 Step 1: Mapping to Trigonometric Space

Mapping a Gaussian distribution $p(\boldsymbol{x})$ to $p(u(\boldsymbol{x})) = p(\sin(a\boldsymbol{x}), \cos(a\boldsymbol{x}))$ does not result in a Gaussian distribution. However, we use a Gaussian approximation since it is convenient for the purpose of long term forecasting. It turns out that the mean and variance of the trigonometrically transformed variable $u(\boldsymbol{x}) \in \mathbb{R}^{2D}$ can be computed analytically. For notational convenience, we will detail the computations in the following for $x \in \mathbb{R}$, but the extension to multivariate inputs $\boldsymbol{x} \in \mathbb{R}^D$ is given in Appendix C.1.

Let us assume that $p(x) = \mathcal{N}(x|\mu, \sigma^2)$. The mean vector $\tilde{\boldsymbol{\mu}}$ and covariance matrix $\tilde{\boldsymbol{\Sigma}}$ of $p(u(x))$ are given as

$$\tilde{\boldsymbol{\mu}} = \begin{bmatrix} \mathbb{E}[\sin(ax)] \\ \mathbb{E}[\cos(ax)] \end{bmatrix}, \tag{3.10}$$

$$\tilde{\boldsymbol{\Sigma}} = \begin{bmatrix} \mathbb{V}[\sin(ax)] & \mathbb{C}[\sin(ax), \cos(ax)] \\ \mathbb{C}[\cos(ax), \sin(ax)] & \mathbb{V}[\cos(ax)] \end{bmatrix}, \tag{3.11}$$

where the covariance between two variables is denoted by $\mathbb{C}$.

Using results from convolving trigonometric functions with Gaussians [13], we obtain

$$\mathbb{E}[\sin(ax)] = \int \sin(ax)p(x)\mathrm{d}x = \exp(-\tfrac{1}{2}a^2\sigma^2)\sin(a\mu), \tag{3.12}$$

$$\mathbb{E}[\cos(ax)] = \int \cos(ax)p(x)\mathrm{d}x = \exp(-\tfrac{1}{2}a^2\sigma^2)\cos(a\mu), \tag{3.13}$$

which allows us to compute the mean $\tilde{\boldsymbol{\mu}}$ in eq. (3.10) analytically.

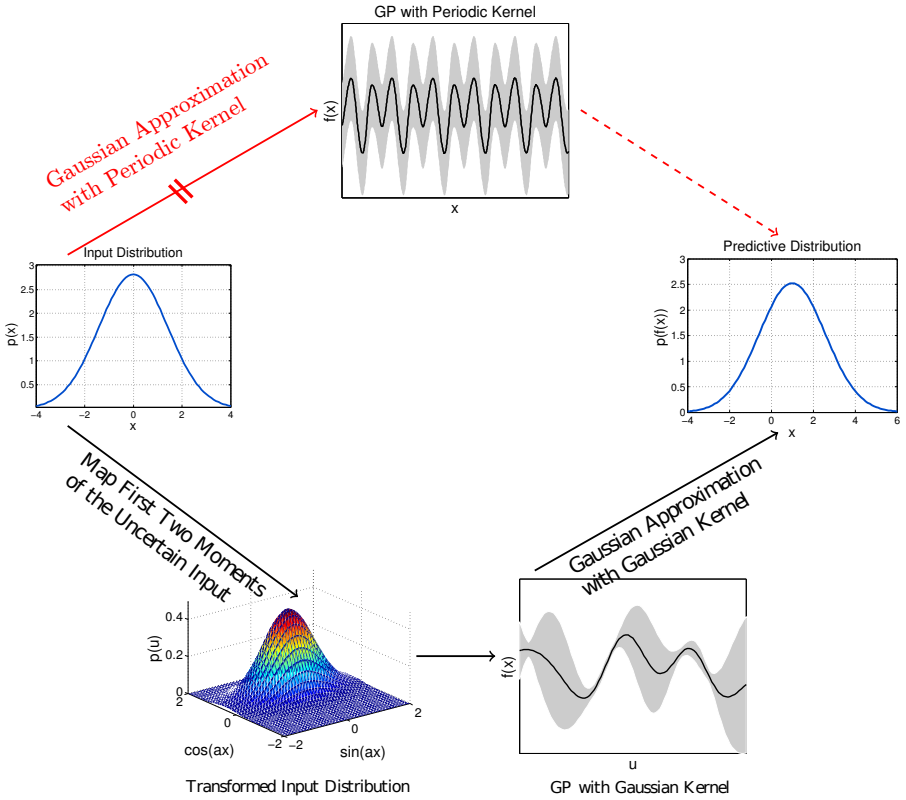


Figure 3.2: The red line on the top shows that there is no analytic solution for the exact moment matching with the standard periodic kernel. Instead, we use the two-step approximation approach to obtain an approximate solution (bottom path). First, the first two moments of the input are mapped analytically to the trigonometric space. Subsequently, the input in the trigonometric space is estimated with the Gaussian approximation with the Gaussian kernel.

To compute the covariance matrix $\tilde{\boldsymbol{\Sigma}}$ in eq. (3.11), we need to compute the variances $\mathbb{V}[\sin(ax)]$, $\mathbb{V}[\cos(ax)]$ and the cross-covariance terms $\mathbb{C}[\sin(ax), \cos(ax)]$.

The variance of $\sin(ax)$ is given by

$$\mathbb{V}[\sin(ax)] = \mathbb{E}[\sin^2(ax)] - \mathbb{E}[\sin(ax)]^2, \tag{3.14}$$

where $\mathbb{E}[\sin(ax)]$ is given in eq. (3.12) and

$$\mathbb{E}[\sin^2(a\boldsymbol{x})] = \int \sin^2(ax)p(x)dx \tag{3.15}$$

$$= \tfrac{1}{2}(1 - \exp(-2a^2\sigma^2)\cos(2a\mu)). \tag{3.16}$$

Similarly, the variance of $\cos(ax)$ is given by

$$\mathbb{V}[\cos(ax)] = \mathbb{E}[\cos^2(ax)] - \mathbb{E}[\cos(ax)]^2, \tag{3.17}$$

where $\mathbb{E}[\cos(ax)]$ is given in eq. (3.13) and

$$\mathbb{E}[\cos^2(ax)] = \tfrac{1}{2}(1 + \exp(-2a^2\sigma^2)\cos(2a\mu)). \tag{3.18}$$

The cross-covariance term $\mathbb{C}[\sin(ax), \cos(ax)]$ is

$$\mathbb{C}[\sin(ax), \cos(ax)] = \mathbb{E}[\sin(ax)\cos(ax)] - \mathbb{E}[\sin(ax)]\mathbb{E}[\cos(ax)], \tag{3.19}$$

where $\mathbb{E}[\sin(ax)]$ and $\mathbb{E}[\cos(ax)]$ are given in eq. (3.12) and (3.13), respectively. The first term in eq. (3.19) is computed according to

$$\mathbb{E}[\sin(ax)\cos(ax)] = \tfrac{1}{2}\exp(-2a^2\sigma^2)\sin(2a\mu), \tag{3.20}$$

where we exploited that $\sin(x)\cos(x) = \sin(2x)/2$.

The results allow us to analytically compute the mean $\tilde{\boldsymbol{\mu}}$ and the covariance matrix $\tilde{\boldsymbol{\Sigma}}$ of a trigonometrically transformed variable $u(\boldsymbol{x})$. In the following, we apply results from [10, 11] to map the trigonometric transformed input through a GP with a Gaussian kernel to compute the mean and the covariance of $p(f(\boldsymbol{x}_*))$.

3.1.4 Step 2: Computing the Predictive Distribution

Now we turn to the second step of the double approximation, which is the analytic computation of the terms in eq. (3.5) and eq. (3.6) with the trigonometrically transformed inputs $u(\boldsymbol{x}_*)$. For this purpose, we also map the GP training inputs $\boldsymbol{X}$ trigonometrically into $\boldsymbol{U}$. Many derivations in the following are based on the work by Deisenroth [14, 12].

The predictive mean $m(\boldsymbol{x}_*)$ in eq. (3.3) can now be written as

$$m(\boldsymbol{x}_*) = \tilde{\boldsymbol{\beta}}^\top \int k_{\mathrm{SE}}(\boldsymbol{U}, \boldsymbol{u}_*)\mathcal{N}(\boldsymbol{u}_*|\tilde{\boldsymbol{\mu}}_*, \tilde{\boldsymbol{\Sigma}}_*)d\boldsymbol{u}_*\,,$$

where we define $\tilde{\boldsymbol{\beta}} = (k_{\mathrm{SE}}(\boldsymbol{U}, \boldsymbol{U}) + \sigma_\varepsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y} \in \mathbb{R}^n$. Note that the kernel in this integral is no longer a periodic kernel, but a Gaussian, applied to the trigonometrically transformed inputs $\boldsymbol{u}_*$. Since this integral is the product of two Gaussian shaped functions it can be solved analytically [11]. We define

$$\boldsymbol{q} = \int k_{\mathrm{SE}}(\boldsymbol{u}_*, \boldsymbol{U})p(\boldsymbol{u}_*)d\boldsymbol{u}_*,$$

where the elements of $\boldsymbol{q} \in \mathbb{R}^n$ are given by

$$q_j = \frac{\alpha^2}{\sqrt{|\tilde{\boldsymbol{\Sigma}}_*\boldsymbol{\Lambda}^{-1}+\boldsymbol{I}|}} \exp(-\tfrac{1}{2}\boldsymbol{\zeta}_j^\top(\tilde{\boldsymbol{\Sigma}} + \boldsymbol{\Lambda})^{-1}\boldsymbol{\zeta}_j), \tag{3.21}$$

for $j = 1, \ldots, n$, where $\boldsymbol{\zeta}_j = (\boldsymbol{u}_j - \tilde{\boldsymbol{\mu}}_*)$.

To compute the predictive covariance $v(\boldsymbol{x}_*)$, we need to solve the following integrals, see equations (3.5)–(3.6):

$$\int k_{\mathrm{SE}}(\boldsymbol{u}_*, \boldsymbol{u}_*)p(\boldsymbol{u}_*)d\boldsymbol{u}_*\,, \tag{3.22}$$

$$\int k_{\mathrm{SE}}(\boldsymbol{U}, \boldsymbol{u}_*)k_{\mathrm{SE}}(\boldsymbol{u}_*, \boldsymbol{U})p(\boldsymbol{u}_*)d\boldsymbol{u}_*\,. \tag{3.23}$$

Note that the second integral in eq. (3.6) can be expressed in terms of eq. (3.23) by using $\boldsymbol{a}^\top\boldsymbol{b} = \mathrm{Tr}(\boldsymbol{b}\boldsymbol{a}^\top)$. Since the Gaussian kernel k_{SE} is stationary, the integral in eq. (3.22) is simply given by the signal variance α^2. The integral in eq. (3.23) results in a matrix $\boldsymbol{Q}$, whose entries are

$$\begin{aligned} Q_{ij} = |2\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{\Sigma}}_* + \boldsymbol{I}|^{-1/2} \times k_{\mathrm{SE}}(\boldsymbol{u}_i, \tilde{\boldsymbol{\mu}}_*)k_{\mathrm{SE}}(\boldsymbol{u}_j, \tilde{\boldsymbol{\mu}}_*) \\ \times \exp(-\tfrac{1}{2}(\boldsymbol{\nu} - \tilde{\boldsymbol{\mu}}_*)^\top(\tfrac{1}{2}\boldsymbol{\Lambda} + \tilde{\boldsymbol{\Sigma}}_*)^{-1}(\boldsymbol{\nu} - \tilde{\boldsymbol{\mu}}_*)) \end{aligned}$$

for $i, j = 1, \ldots, n$ and with $\boldsymbol{\nu} = (\boldsymbol{u}_i + \boldsymbol{u}_j)/2$.

These results allow us to analytically compute approximations to the predictive distribution for Gaussian processes with periodic kernels. Although all computations can be performed analytically, the additional Gaussian approximation of the trigonometrically transformed state variable $\boldsymbol{u}_*$ (Step 1) makes the computation of predictive mean and variance only approximate.

Chapter 4

Experiments

In this chapter, we shed light on the performance of our proposed approximation method. We present an empirical evaluation of the double approximation. The comparison of the periodic with the Gaussian kernel for the long term predictions is also presented. The experiments are evaluated on different synthetic data sets.

For simulation, the gpml toolbox[1] is used. The toolbox [15] is the MATLAB implementation of the inference and prediction with Gaussian processes. A numerical optimizer is already implemented, which is used for training the GP models. The optimizer maximizes the log marginal likelihood discussed in Section 2.2. We need to add two parts to the toolbox in order to perform our experiments:

1. The periodic kernel in eq. (3.7), as well as the first order derivatives of the function with respect to its parameters (signal variance, periodicity and length-scale parameters) for evidence maximization (see Section 2.2). The derivatives are given in Appendix B.

2. The double approximation method discussed in Chapter 3, including the mapping to the trigonometric space as well as the moment matching with Gaussian kernel.

Both parts were implemented in MATLAB.

4.1 Evaluation of Double Approximation for One-step Prediction

Here, we investigate how the double approximation method performs when applied to a given periodic signal. In Chapter 3, we discussed that the true predictive distribution at an uncertain input is not a Gaussian. We adopt our

[1]The gpml toolbox is publicly available at http://www.gaussianprocess.org/gpml.

proposed double approximation method to approximate the non-Gaussian predictive distribution with a Gaussian for periodic Gaussian processes.

We present the numerical evaluation of the double approximation method on a synthetic data set. Numerical methods rely on sampling techniques that evaluate the intractable integrals numerically. We need to sample from the Gaussian distributed test inputs $\boldsymbol{x}_*$. These samples are deterministic inputs whose predictive outputs are normally distributed. Hence, prediction at these samples can be done analytically, see Section 2.3. As the number of samples grows the approximate distribution will tend to the true distribution [7]. We approximate the resulted sampling distribution by a Gaussian. Finally, the mean and variance of the sampling distribution are compared with the mean and variance obtained by applying the double approximation method.

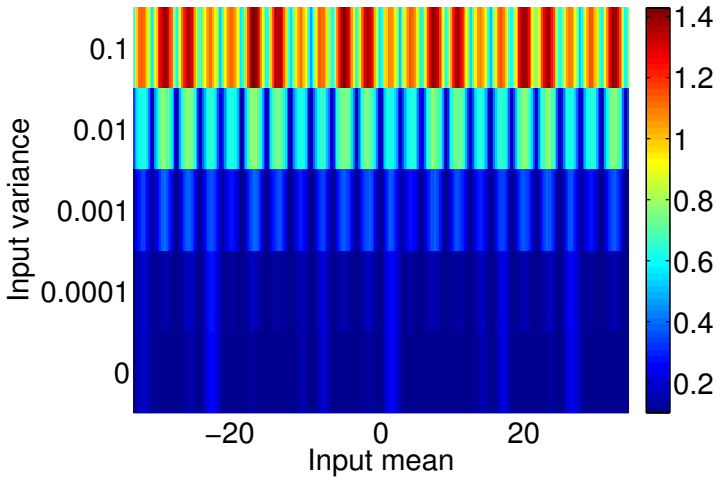
Having this method of evaluation, we consider the system $y = \sin(x/2) + \cos(x + 0.35) + \varepsilon$, where the system's noise is $\varepsilon \sim \mathcal{N}(0, 1.6 \times 10^{-3})$. The GP model with periodic kernel is trained by the evidence maximization method, see Section 2.2. The training set is of size 400, where the training inputs x_i are linearly spaced between -17 and 17. The test data points are in the range $[-11\pi, 11\pi]$. The function and the range of the training data are visualized in Figure 2.4 in blue and red, respectively.

We define test input distributions $p(x_0^{ij}) = \mathcal{N}(\mu_i, \sigma_j^2)$ from which we draw 100 samples x_* at random and map them through the periodic function. Then, we compute the root-mean-square error (RMSE) in eq. (4.1) and the negative log predictive distribution (NLPD) in eq. (4.2) of the true function values, evaluated by the sampling method, with respect to the predictive distributions.

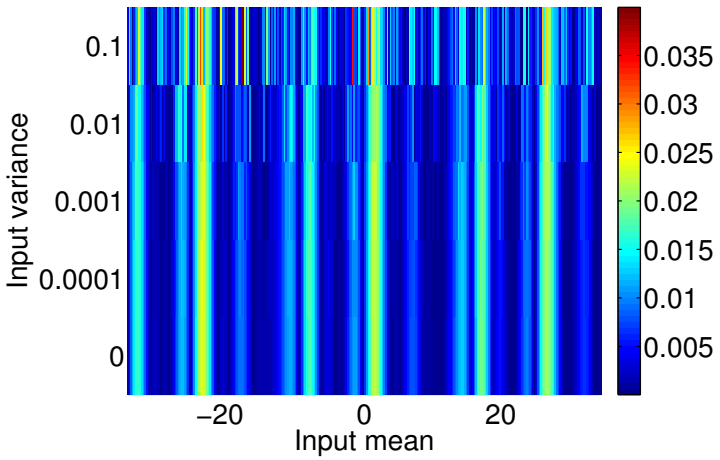
$$\mathrm{RMSE}_{x_*} = \sqrt{\mathbb{E}[(y_s - \mu_{x_*})^2]}, \tag{4.1}$$

$$\mathrm{NLPD}_{x_*} = \frac{1}{2}\log|\Sigma_{x_*}| + \frac{1}{2}(y_s - \mu_{x_*})^\top(\Sigma_{x_*})^{-1}(y_s - \mu_{x_*}) + \frac{D}{2}\log(2\pi). \tag{4.2}$$

While the RMSE only considers the error on means, the NLPD takes the variances into account as well. The mean values μ_i of the test input distributions $p(x_0^{ij})$ are selected on a linear grid from -11π to 11π. The corresponding variances σ_j^2 are set to 10^{-j}, $j = 1, \ldots, 4$. Moreover, we test the approximation for $\sigma_0^2 = 0$, which corresponds to a deterministic input.



(a) NLPD.



(b) RMSE.

Figure 4.1: Quality of the double approximation for the periodic function shown in Figure 2.4. The average NLPD and RMSE values are given for various input distributions whose means and variances are displayed on the horizontal and vertical axes, respectively. Higher variance increases the errors especially the NLPD error, but varying the mean does not have an impact on the errors.

Figure 4.1 displays the RMSE and NLPD values for predictions with the proposed double approximation. It can be seen that the NLPD values are relatively equal for all input variances, see Figure 4.1a. The periodic pattern of the function can be recognized in each row of Figure 4.1a: The predictions were particularly accurate in the linear regimes of the function. The average RMSE values in Figure 4.1b are generally small and do not differ substantially as a function of the variance σ_j^2.

Table 4.1: Average quality of the double approximation.

σ_j^2	0	10^{-4}	10^{-3}	10^{-2}	10^{-1}
NLPD	0.12	0.14	0.24	0.52	1.00
RMSE ($\times 10^{-4}$)	6.5	6.5	6.5	6.7	8.0

Table 4.1 shows the average performance of the double approximation, where we average the NLPD and RMSE values over all means μ_i of the test input distributions. The RMSE values are relatively constant over varying input variances σ_j^2. This means that the mean estimate by the double approximation is relatively robust. The NLPD values on the other hand indicate that the coherence of the predictive variance suffers to from increasing uncertainty in the input distribution.

4.2 Evaluation of Double Approximation for Long Term Forecasting

We evaluate the performance of the Double approximation method for long term forecasting. The experiment is the simulation of a pendulum motion, shown in Figure 4.2. The state of the system $\boldsymbol{x}$ is given by the pair of angle and angular velocity $(\varphi, \dot{\varphi})$. φ is the angle of deviation of pendulum from the vertical at a given moment, measured anti-clockwise in radians. For more details regarding the physical properties of the motion we refer to Appendix D. A constant force was applied to the pendulum, such that it reached a limit-cycle behavior after about 2 s, in which both the angle and the angular velocity followed a periodic pattern. We trained a GP on 300 data points, where the measurement noise variance was $10^{-2}\boldsymbol{I}$.

For model learning, we train the hyper-parameters of the periodic GP (2 periodicity a, 2 length-scales l_i, signal variance α^2, and noise variance σ_ε^2). Moreover, we train a GP with the Gaussian kernel, where the hyper-parameters were two length-scales $\{l_1^2, l_2^2\}$, the signal variance α^2, and the noise variance σ_ε^2. The training targets for both GP models are the differences between consecutive states, i.e., $\boldsymbol{y}_i = \boldsymbol{x}_i - \boldsymbol{x}_{i-1}$, which effectively encodes a linear prior mean function $m(\boldsymbol{x}) = \boldsymbol{x}$. Both GP models are trained by maximizing the marginal likelihood (evidence), see eq. (2.6).

To evaluate the performance of the models for long term forecasting, the models are used to predict the pendulum's state evolution for $T = 100$ time steps ahead. We set the initial covariance to $0.01\boldsymbol{I}$. For long term forecasting with periodic kernel, we repeat the double approximation (see Chapter 3) for T times, where the output at each state serves as input for the successor state $p(\boldsymbol{x}_{t+l+1}|\boldsymbol{x}_{t+l})$. In this setting, not only the mean is computed itera-

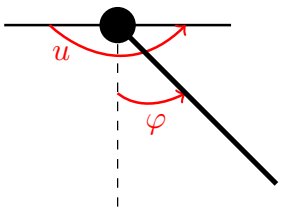


Figure 4.2: Pendulum.

tively but also the uncertainty is propagated through the time. It is worth to say that, for the Gaussian approximation with the Gaussian kernel, we follow a similar method as discussed in Chapter 3 with one difference. There is no need to map the input to the trigonometric space, since that step is done for encoding the periodicity into the model.

Figure 4.3 illustrates the result of the experiment. The first and second rows show the result for the angle and the angular velocity, respectively. The left column shows the result for the Gaussian kernel, while the right column illustrates the results for the periodic kernel. The error bars are shown by the blue vertical lines, corresponding to the mean plus and minus two times the standard deviation. The small error on the training set indicates that both kernels can predict well where the training data is available. For the test set, however, the Gaussian kernel loses track of the data. In contrast, the GP with the periodic kernel can predict the test points successfully up to the time horizon T.

We also present the NLPD and RMSE errors on long term forecasting of the pendulum motion. To have statistically meaningful results, the experiment described above is repeated for 100 starting points drawn randomly from 600 test points. From each of these starting points, we perform long term forecasting up to the time horizon of 100.

Figure 4.4 illustrates the average NLPD error for 100 steps. This result is in alignment with what we observed previously. In Figure 4.3, as the number of steps increases, the variance and difference between the predictive means and the function values increase and as a result errors are growing. The error has the same inclining trend for GPs with periodic and Gaussian kernel. However, the error of the periodic GP is consistently smaller than GP with the Gaussian kernel.

The RMSE error is illustrated in Figure 4.5. Since there is no direct way to generalize the RMSE to the multivariate case, the errors on two features are computed separately. As we mentioned before, the only factor

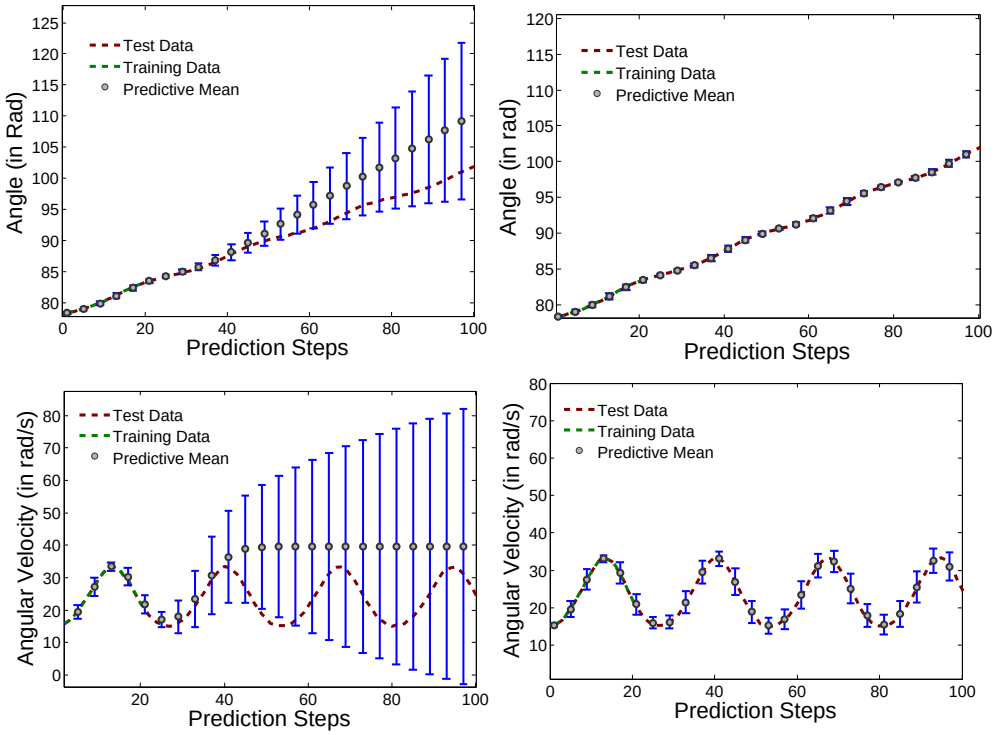
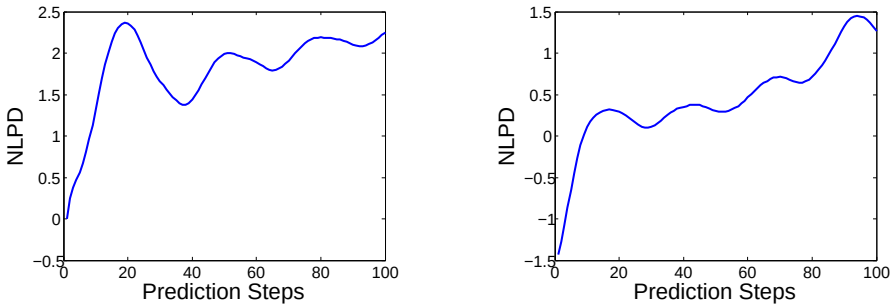


Figure 4.3: The x-axes represent the time, the y-axes represent the angle of the pendulum at each time, in the top figures and the angular velocity, in the bottom figures. The right and left columns illustrate the Gaussian and the periodic kernel, respectively. T here is set to 100, which means we concatenate one-step ahead prediction 100 times. The blue lines represents the predictive mean plus and minus two times the standard deviation.

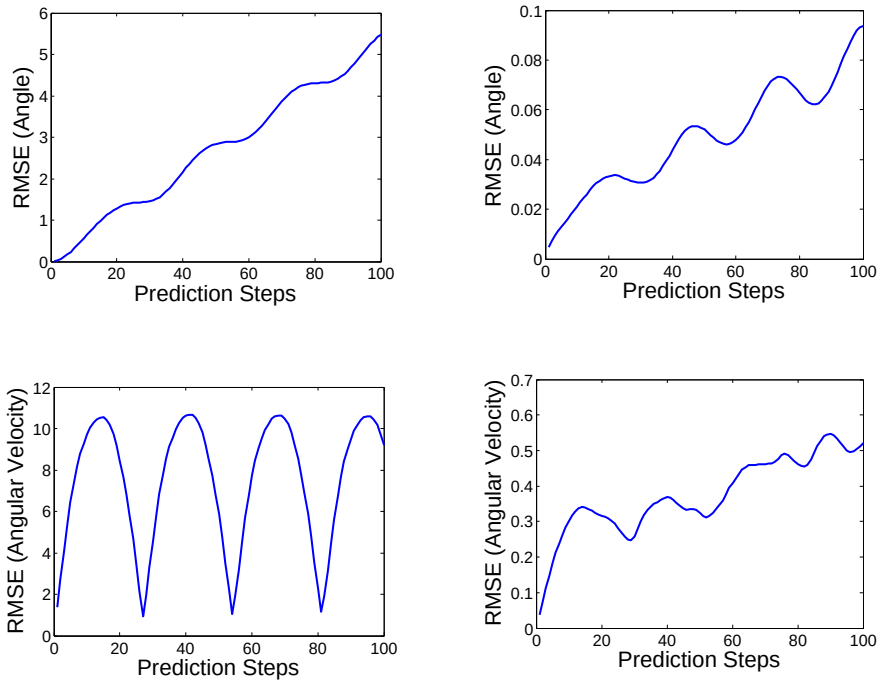
that plays a role here is the difference between the predictive means and the function values, see eq. (4.1). Note how this difference is changing with respect to the time steps for each feature in Figure 4.3. For the periodic GP, the RMSE slightly increases for both features, Figure 4.5 (right panels). The lower-left panel in Figure 4.3 shows prediction of the angular velocity with GP with the Gaussian kernel. The predictive distributions of the test data are constant with respect to the time steps. Hence, the error is the root-mean-square of the difference between a constant value and a periodic signal, which result to what illustrated in Figure 4.5 (lower-left panel). The figure demonstrates that for both features, the periodic GP performs significantly better than the GP with the Gaussian kernel.

Generally, the results show that the Gaussian kernel can make accurate prediction in areas that the training inputs are provided. But it fails to extrapolate from the training set to the test set. The experiments confirm that the periodic GPs successfully extracts the periodic pattern of the underlying function and generalizes to the new test data points. Both NLPD and RMSE errors are small for the periodic GP for such a long time horizon of 100 steps, which indicates that the double approximation is a rewarding method for long term forecasting of periodic systems.



(a) NLPD for GP with Gaussian kernel (b) NLPD for GP with periodic kernel

Figure 4.4: The figure demonstrates the error on the long term prediction of the pendulum motion. The left and right panel illustrate the negative log predictive distribution error for a GP with the Gaussian and the periodic kernel, respectively. Both errors grow as the steps increase. Although, the error of the periodic GP is consistently smaller than the error of GP with the Gaussian kernel.



(a) RMSE for GP with Gaussian kernel (b) RMSE for GP with periodic kernel

Figure 4.5: RMSE error is shown for two features of angle and angular velocity for the periodic GP and GP with the Gaussian kernel. Smaller errors for the periodic GP (right panels) proves that it outperforms the GP with the Gaussian kernel (left panels) for long term forecasting of periodic signals.

Chapter 5

Conclusion

We have discussed long term forecasting of periodic systems using Gaussian processes. For long term forecasting, we have iteratively computed predictive distributions up to a time horizon T. It is necessary to propagate uncertainty associated with the prediction at each state to the successor states. In such a setting, we need to predict at uncertain inputs, which is analytically intractable with the periodic GP models. We have seen that the moment matching method allows analytic prediction at Gaussian distributed inputs. However, analytic moment matching is only possible for some kernels, such as Gaussian and polynomial kernels. In case of the standard periodic kernel which is of interest in our work, long term forecasting with analytic moment matching is intractable.

We have proposed an equivalent parametric form of the standard periodic kernel, which, in combination with a double approximation method, allows for long term forecasting of periodic processes. At the first step of the double approximation, the first two moments of the input distribution have been mapped to the trigonometric space, to embed the periodicity property of the underlying function into the model. Subsequently, we have mapped the trigonometrically transformed input through the GP function with the Gaussian kernel. Both steps have been an analytic approximation of a non-Gaussian distribution to a Gaussian.

Furthermore, the empirical evaluation of the double approximation has been presented. To answer the first research question regarding the robustness of the double approximation against varying the test input distribution, we examined it on a periodic example system, see Section 4.1. The results indicate that the method is robust against varying the mean of the test inputs but it suffers to some extent from increasing variance of the input distribution, see Table 4.1.

The next part of our experiments has provided a comparison of the periodic kernel with the Gaussian kernel for long term forecasting of a periodic system (second research question). The results show that non-periodic kernels such as the Gaussian kernel fail to perform long term forecasting while the periodic kernel have brought promising result in this regard, see Section 4.2. It indicates that using a periodic kernel for long term prediction of the periodic systems is essential.

The last part of our experiments answers the last research question regarding the performance of the double approximation on long term forecasting, see Section 4.2. The results confirm that as the time steps increase the prediction error grows. Nevertheless, errors stay small, which implies that our double approximation method is a valuable method for long term forecasting of the periodic Gaussian processes.

5.1 Future Work

In this work we examined the systems with exact periodic patterns. Real world applications, however, may not always follow exact periodic patterns. For instance, other trends may exist in systems in addition to the periodic trend. Such situations may hinder modeling and prediction with periodic Gaussian processes. In future, we will generalize our inference method to the signals that are not exactly periodic. This can be achieved by multiplying the periodic kernel with a Gaussian kernel, for instance, which has also been suggested by Roberts et al. [16]. We will investigate the extension of these models to long term forecasting.

Appendix A

Mathematical Tools

A.1 Gaussian Identities

Let $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{a}, \boldsymbol{A})$ be a multivariate normally distributed random variable with mean vector $\boldsymbol{a}$ and the covariance matrix $\boldsymbol{A}$ with the size of $D \times 1$ and $D \times D$ respectively. Then the Gaussian density becomes

$$p(x) = (2\pi)^{-\frac{D}{2}} |\boldsymbol{A}|^{-\frac{1}{2}} \exp[-\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{a})^\top \boldsymbol{A}^{-1}(\boldsymbol{x} - \boldsymbol{a})]. \tag{A.1}$$

A.1.1 Marginal and Conditional Distributions

Let $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{B})$. Then the joint normal distribution $p(\boldsymbol{x}, \boldsymbol{y})$ becomes

$$p(\boldsymbol{x}, \boldsymbol{y}) = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}, \begin{bmatrix} \boldsymbol{A} & \boldsymbol{C} \\ \boldsymbol{C}^\top & \boldsymbol{B} \end{bmatrix} \right), \tag{A.2}$$

where $\boldsymbol{C}$ is the matrix of cross-covariances between $\boldsymbol{x}$ and $\boldsymbol{y}$.

The marginal distributions are then

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{a}, \boldsymbol{A}), \tag{A.3}$$
$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{b}, \boldsymbol{B}), \tag{A.4}$$

and the conditional distributions

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{a} + \boldsymbol{C}\boldsymbol{B}^{-1}(\boldsymbol{y} - \boldsymbol{b}), \boldsymbol{A} - \boldsymbol{C}\boldsymbol{B}^{-1}\boldsymbol{C}^\top), \tag{A.5}$$
$$p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{b} + \boldsymbol{C}^\top \boldsymbol{A}^{-1}(\boldsymbol{x} - \boldsymbol{a}), \boldsymbol{B} - \boldsymbol{C}^\top \boldsymbol{A}^{-1}\boldsymbol{C}). \tag{A.6}$$

A.1.2 Product of Gaussians

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{A})\mathcal{N}(\boldsymbol{y}|\boldsymbol{b},\boldsymbol{B}) = c\mathcal{N}(\boldsymbol{c},\boldsymbol{C}) \tag{A.7}$$

$$\boldsymbol{c} = \boldsymbol{C}(\boldsymbol{A}^{-1}\boldsymbol{a} + \boldsymbol{B}^{-1}\boldsymbol{b}), \boldsymbol{C} = (\boldsymbol{A}^{-1} + \boldsymbol{B}^{-1})^{-1}, \tag{A.8}$$

and with the constant c

$$c = (2\pi)^{-\frac{1}{2}}|\boldsymbol{A} + \boldsymbol{B}|^{-\frac{1}{2}} \exp(-\tfrac{1}{2}(\boldsymbol{a} - \boldsymbol{b})^\top (\boldsymbol{A} + \boldsymbol{B})^{-1}(\boldsymbol{a} - \boldsymbol{b})) \tag{A.9}$$

A.1.3 Matrix Derivatives

Derivative of an inverse matrix [17] becomes

$$\frac{\partial}{\partial \boldsymbol{\theta}}\boldsymbol{A}^{-1} = -\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{\theta}}\boldsymbol{A}^{-1}, \tag{A.10}$$

where $\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{\theta}}$ is a matrix of element-wise derivatives. The derivative of the log determinant of a positive definite symmetric matrix $\boldsymbol{A}$ is given by

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log |\boldsymbol{A}| = \mathrm{Tr}(\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{\theta}}). \tag{A.11}$$

A.2 Law of Iterated Expectations

Suppose x and y are two random variables. The law of iterated expectations can be expressed as

$$\mathbb{E}[\mathbb{E}[x|y]] = \mathbb{E}[x], \tag{A.12}$$

A.3 Trigonometric Identities

Here we present some of trigonometric identities [13] that are used in our computations

$$\cos(2x) = 1 - 2\sin(x)^2, \tag{A.13}$$
$$\sin(x \pm y) = \sin(x)\cos(y) \pm \cos(x)\sin(y), \tag{A.14}$$
$$\cos(x \pm y) = \cos(x)\cos(y) \mp \sin(x)\sin(y), \tag{A.15}$$
$$\sin(x)\sin(y) = \tfrac{1}{2}[\cos(x - y) - \cos(x + y)], \tag{A.16}$$
$$\cos(x)\cos(y) = \tfrac{1}{2}[\cos(x - y) + \cos(x + y)], \tag{A.17}$$
$$\sin(x)\cos(y) = \tfrac{1}{2}[\sin(x + y) + \sin(x - y)]. \tag{A.18}$$

Appendix B

Derivatives of Periodic Kernel

In this appendix, we present the derivatives of the proposed periodic function in eq. (3.7) w.r.t. its hyper-parameters. For the input $\boldsymbol{x} \in \mathbb{R}^D$ the periodic kernel becomes

$$\boldsymbol{K} = k_{per}(\boldsymbol{x}, \boldsymbol{x}') = \alpha^2 \exp\left(-\tfrac{1}{2}\sum_{d=1}^{D} \boldsymbol{z}_d^\top \boldsymbol{\Lambda}_d^{-1} \boldsymbol{z}_d\right), \tag{B.1}$$

where we have $\boldsymbol{z}_d$ as the d^{th} dimension of the trigonometrically transformed input and $\boldsymbol{\Lambda}_d = \text{diag}[l^{d^2}, l^{d^2}]$. Moreover, $\boldsymbol{K}$ is the covariance matrix and the hyper-parameter vector is $\boldsymbol{\theta} = (\alpha^2, \boldsymbol{l}, \boldsymbol{a})$. The derivative of a matrix $\boldsymbol{K}$ can be broken to the derivatives of the elements of the matrix K_{ij}. Hence, the first order derivative of K_{ij} w.r.t. the α^2 is given by

$$\frac{\partial K_{ij}}{\partial \alpha^2} = 2\alpha \times K_{ij}. \tag{B.2}$$

The first derivative w.r.t. the length-scales $\{l_d\}_{d=1}^D$ and periodicity hyper-parameters $\{a_d\}_{d=1}^D$ are $D \times 1$ vectors whose d^{th} component can be written as

$$\begin{aligned} \frac{\partial K_{ij}}{\partial a^d} &= \frac{\partial}{\partial a^d}\left[-\tfrac{1}{2}\boldsymbol{z}_d^\top \boldsymbol{\Lambda}_d^{-1} \boldsymbol{z}_d\right] K_{ij} \\ &= \Big(\sin(a^d x_i^d)\cos(a^d x_j^d) - \sin(a^d x_j^d)\cos(a^d x_i^d)\Big)\frac{(x_i^d - x_j^d)}{l^{d^2}} K_{ij}, \end{aligned} \tag{B.3}$$

and

$$\begin{aligned} \frac{\partial K_{ij}}{\partial l^d} &= \frac{\partial}{\partial l^d}\left[-\tfrac{1}{2}\boldsymbol{z}_d^\top \boldsymbol{\Lambda}_d^{-1} \boldsymbol{z}_d\right] K_{ij} \\ &= \left(-\boldsymbol{\Lambda}_d^{-T} \boldsymbol{z}_d \boldsymbol{z}_d^\top \boldsymbol{\Lambda}_d^{-T}\right) K_{ij}. \end{aligned} \tag{B.4}$$

Appendix C

Appendix to Chapter 3

C.1 Mapping to Trigonometric Space for Multivariate Input

In Section 3.1.3, we discussed mapping the one dimensional inputs to the trigonometric space. Here we discuss such a mapping for multivariate inputs $\boldsymbol{x} \in \mathbb{R}^D$. Assume that $\boldsymbol{x}$ is a Gaussian distribution. We map its first two moments to the $u = (\sin(\boldsymbol{a}\boldsymbol{x}), \cos(\boldsymbol{a}\boldsymbol{x}))$. Note that we can learn a periodicity hyper-parameter for each data dimension. As a result, the trigonometrically transformed mean $\tilde{\boldsymbol{m}}$ and covariance matrix $\tilde{\boldsymbol{\Sigma}}$ becomes

$$\tilde{\boldsymbol{m}} = \begin{bmatrix} \mathbb{E}[\sin(a_1 x_1)] \\ \mathbb{E}[\cos(a_1 x_1)] \\ \vdots \\ \mathbb{E}[\sin(a_D x_D)] \\ \mathbb{E}[\cos(a_D x_D)] \end{bmatrix}, \tag{C.1}$$

and

$$\tilde{\boldsymbol{\Sigma}} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_{11} & \tilde{\boldsymbol{\Sigma}}_{12} & \dots & \tilde{\boldsymbol{\Sigma}}_{1D} \\ \vdots & \ddots & \ddots & \vdots \\ \tilde{\boldsymbol{\Sigma}}_{D1} & \dots & \dots & \tilde{\boldsymbol{\Sigma}}_{DD} \end{bmatrix}, \tag{C.2}$$

where for two dimensions of i and j, if $i = j$

$$\tilde{\boldsymbol{\Sigma}}_{ii} = \begin{bmatrix} \mathbb{V}[\sin(a_i x_i)] & \mathbb{C}[\sin(a_i x_i), \cos(a_i x_i)] \\ \mathbb{C}[\cos(a_i x_i), \sin(a_i x_i)] & \mathbb{V}[\cos(a_i x_i)] \end{bmatrix}, \tag{C.3}$$

and if $i \neq j$

$$\tilde{\boldsymbol{\Sigma}}_{ij} = \begin{bmatrix} \mathbb{C}[\sin(a_i x_i), \sin(a_j x_j)] & \mathbb{C}[\sin(a_i x_i), \cos(a_j x_j)] \\ \mathbb{C}[\cos(a_i x_i), \sin(a_j x_j)] & \mathbb{C}[\cos(a_i x_i), \cos(a_j x_j)] \end{bmatrix}, \tag{C.4}$$

where $\mathbb{E}$ is the expected value, $\mathbb{V}$ is the variance, and $\mathbb{C}$ is the covariance function. In Section 3.1.3, computation of the elements of matrix $\tilde{\mathbf{\Sigma}}_{ij}$ is presented for $i = j$. Here, we present the computation when $i \neq j$.

$$\begin{aligned} &\mathbb{C}[\sin(a_i x_i), \sin(a_j x_j)] \\ &= \mathbb{E}[\sin(a_i x_i)\sin(a_j x_j)] - \mathbb{E}[\sin(a_i x_i)]\mathbb{E}[\sin(a_j x_j)], \end{aligned} \tag{C.5}$$

where both $\mathbb{E}[\sin(a_i x_i)]$ and $\mathbb{E}[\sin(a_j x_j)]$ can be computed based on eq. (3.12). Using $\sin(x)\sin(y) = \frac{1}{2}[\cos(x-y) - \cos(x+y)]$, the first term in eq. (C.5) becomes

$$\begin{aligned} &\mathbb{E}[\sin(a_i x_i)\sin(a_j x_j)] \\ &= \frac{1}{2}\mathbb{E}[\cos(a_i x_i - a_j x_j) - \cos(a_i x_i + a_j x_j)] \\ &= \frac{1}{2}\left(\mathbb{E}[\cos(a_i x_i - a_j x_j)] - \mathbb{E}[\cos(a_i x_i + a_j x_j)]\right). \end{aligned} \tag{C.6}$$

We have $x_i \sim \mathcal{N}(\mu_i, \sigma_{ii}^2)$, $x_j \sim \mathcal{N}(\mu_j, \sigma_{jj}^2)$ and $\mathbb{C}(x_i, x_j) = \sigma_{ij}^2$, then

$$p(a_i x_i \pm a_j x_j) = \mathcal{N}(\mathbb{E}[a_i x_i \pm a_j x_j], \mathbb{C}(a_i x_i \pm a_j x_j)), \tag{C.7}$$

where mean and covariance are given by

$$\mathbb{E}[a_i x_i \pm a_j x_j] = a_i \mathbb{E}[x_i] \pm a_j \mathbb{E}[x_j] = a_i \mu_i \pm a_j \mu_j, \tag{C.8}$$

$$\begin{aligned} \mathbb{C}(a_i x_i \pm a_j x_j) &= \mathbb{V}(a_i x_i) + \mathbb{V}(a_j x_j) \pm 2\mathbb{C}(a_i x_i, a_j x_j) \\ &= a_i^2 \sigma_{ii}^2 + a_j^2 \sigma_{jj}^2 \pm 2 a_i a_j \sigma_{ij}^2. \end{aligned} \tag{C.9}$$

From eq. (3.13), we have $\mathbb{E}[\cos(x)] = \exp(-\frac{1}{2}\sigma^2)\cos(\mu)$. Now by substituting the μ by $\mathbb{E}[a_i x_i \pm a_j x_j]$ and σ^2 by $\mathbb{C}(a_i x_i \pm a_j x_j)$, we can compute the two terms of $\mathbb{E}[\cos(a_i x_i \pm a_j x_j)]$ in eq. (C.6).

Other terms in the matrix $\tilde{\mathbf{\Sigma}}_{ij}$ can be solved with the same fashion. First, we change multiplication with a sum by using the trigonometric identities given in equations (A.16) – (A.18). Then, the cross-covariances can be computed based on the equations (3.12) – (3.19) with the means and the variances on the equations (C.8) and (C.9).

Appendix D

Equations of Motion for Pendulum

A pendulum shown in Figure 4.2 is a mass hung from a fixed point so that it can swing freely backward and forward. Typical values for the weight of pendulum m as well as its length l are $1kg$ and $1m$ respectively. φ is the angle of deviation of pendulum from the vertical at a given moment. The Cartesian coordinates of the fixed point are

$$x = \frac{1}{2} l \sin(\varphi), \tag{D.1}$$

$$y = \frac{1}{2} l \cos(\varphi). \tag{D.2}$$

The velocity is the derivative of the position vector with respect to time. The squared velocity of pendulum is

$$v^2 = \frac{1}{4} l^2 \dot{\varphi}, \tag{D.3}$$

where $\dot{\varphi}$ denotes the angular velocity which is the derivation of angle φ with respect to time.

There are two forces; gravitation g which is a downwards force, and also an external force u. Considering the angle and angular velocity $z = [\dot{\varphi}, \varphi]^\top$, the motion of pendulum can be formulated as two ordinary differential equations

$$\frac{\partial z}{\partial t} = \begin{bmatrix} \frac{u - b\varphi - \frac{1}{2} mlg \sin \varphi}{\frac{1}{4} ml^2 + I} \\ \dot{\varphi} \end{bmatrix}, \tag{D.4}$$

where b denotes friction coefficient in eq. (D.4).

The true periodicity T of the angle is determined according to eq. (D.5).

$$T = 2\pi\sqrt{\frac{l}{g}}. \tag{D.5}$$

Bibliography

[1] A. O'Hagan. On curve fitting and optimal design for regression (with discussion). *Journal of the Royal Statistical Society B*, 40(1):142, 1978.

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

[3] B. Schölkopf and A. J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.

[4] N. Durrande, J. Hensman, M. Rattray, and N. D. Lawrence. Gaussian process models for periodicity detection. http://arxiv.org/abs/1303.7090, 2013.

[5] S. Reece and S. Roberts. The near constant acceleration gaussian process kernel for tracking. *IEEE Signal Processing Letters*, 8(17):707–710, 2010.

[6] D. J. C. MacKay. Introduction to Gaussian processes. *Neural Networks and Machine Learning*, 1998.

[7] R. M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical report, Technical Report CRG-TR-92-1, Department of Computer Science, University of Toronto, 1993.

[8] J. Quinonero-Candela, A. Girard, J. Larsen, and C. E. Rasmussen. Propagation of uncertainty in Bayesian kernel models. In *ICASSP*, pages 701–704, 2003.

[9] C. M. Bishop. *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[10] M. P. Deisenroth. *Efficient reinforcement learning using Gaussian processes*. PhD thesis, KIT scientific Publishing, 2010.

[11] A. Girard, C. E. Rasmussen, J. Quinonero-Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs– Application to multiple-step ahead time series forecasting. In *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

[12] M. Deisenroth, D. Fox, and C. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2013.

[13] I. S. Gradshteyn and I. M. Ryzhik. *Table of integrals, series, and products*. Academic Press, 6th edition, 2000.

[14] M. P. Deisenroth and C. E. Rasmussen. Pilco:a model-based and data-efficient approach to policy search. In *ICML*, 2014.

[15] C. E. Rasmussen and H. Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11:3011–3015, 2010.

[16] S. Roberts, M. A. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time series modelling. *Philosophical Transactions of the Royal Society (Part A)*, 371(1984), February 2013.

[17] K. B. Petersen and M. S. Pedersen. *The matrix cookbook*. Academic Press, 20121115 edition, 2008.