Nonparametric Bayesian Mixed-effects Models for Multi-task Learning

A dissertation submitted by

Yuyang Wang

B.M., Beijing Information Technology Institute (2005)
M.S., Tsinghua University (2008)

In partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Computer Science

TUFTS UNIVERSITY

August, 2013

Adviser: RONI KHARDON

Abstract

In many real world problems we are interested in learning multiple tasks while the training set for each task is quite small. When the different tasks are related, one can learn all tasks simultaneously and aim to get improved predictive performance by taking advantage of the common aspects of all tasks. This general idea is known as *multi-task learning* and it has been successfully investigated in several technical settings, with applications in many areas.

In this thesis we explore a Bayesian realization of this idea especially using Gaussian Processes (GP) where sharing the prior and its parameters among the tasks can be seen to implement multi-task learning. Our focus is on the functional mixed-effects model. More specifically, we propose a family of novel Nonparametric Bayesian models, *Grouped mixed-effects GP* models, where each individual task is given by a fixed-effect, taken from one of a set of unknown groups, plus a random individual effect function that captures variations among individuals. The proposed models provide a unified algorithmic framework to solve time series prediction, clustering and classification.

We propose the shift-invariant version of *Grouped mixed-effects GP* to cope with periodic time series that arise in astrophysics when using data for periodic variable stars. We develop an efficient EM algorithm to learn the parameters of the model, and as a special case we obtain the Gaussian mixture model and EM algorithm for phased-shifted periodic time series. Furthermore, we extend the proposed model by using a Dirichlet Process prior, thereby leading to an infinite mixture model. A Variational Bayesian approach is developed for inference in this model, leading to an efficient algorithm for model selection that automatically chooses an appropri-

ate model order for the data.

We present the first sparse solution to learn the *Grouped mixed-effects GP*. We show that, given a desired model order, how the sparse approximation can be obtained by maximizing a variational lower bound on the marginal likelihood, generalizing ideas from single-task Gaussian processes to handle the mixed-effects model as well as grouping.

Finally, the thesis investigates the period estimation problem through the lens of machine learning. Using GP, we propose a novel method for period finding that does not make assumptions on the shape of the periodic function. The algorithm combines gradient optimization with grid search and incorporates several mechanisms to overcome the high computational complexity of GP. We also propose a novel approach for using domain knowledge, in the form of a probabilistic generative model, and incorporate such knowledge into the period estimation algorithm, yielding significant improvements in the accuracy of period identification.

Dedicated to Mom and Dad.

Acknowledgments

No words could ever express my admiration and gratitude to my adviser, Roni Khardon. He is a true teacher, mentor, collaborator and role model. Roni has always been there to guide me, assist me and help me in every possible way, but never "impose his views". Especially during my early PhD days, Roni was as patient as a human being can ever be, tolerating my immature, sloppiness and lack of concentration, never giving up on me and pointing me to the right direction over and over again. For all these, a proper thank you weighs far more than this thesis.

I am extremely fortunate to meet and work with Alan Edelman, who shows me the beauty of mathematics, who teaches me another dimension of thinking and even how to program. I will cherish the fun moment at MIT 2-343 where we explored the random matrix jungle. I will never forget his saying "there are no such thing as hard mathematics, only fun mathematics". While I am forever indebted to Alan for introducing me to the fascinating world of random matrix theory, yet I value his friendship above all. I wish you, Susan, Daniel and Jonnie all the best!

I am greatly indebted to my Committee members, Carla Brodley, Anselm Blumer, Shuchin Aeron and Stan Sclaroff, for their time and valuable feedback. Carla has always been there for me throughout my PhD study. I have also greatly benefitted from interesting conversations with Anselm and Shuchin. Finally, I am really honored to have had Stan on my committee.

Most of the work in this thesis is a result of a collaboration with Pavlos Protopapas from the Harvard Astrophysics department. I would like to thank Pavlos for teaching my astrophysics and making the astrophysical data sets easily accessible

(at all levels) to me.

I like to thank the wonderful staff, faculty members at Tufts CS department. In particular, thanks to Lenore Cowen, Judith Stafford, Jeannine Vangelist, Donna Cirelli and Gail Fitzgerald. I would love to thank all members (past and current) at Tufts ML group, in particular Gabe Wachman, Saket Joshi, Umma Rebbapragada, D. Sculley, Saeed Majidi, and Bilal Ahmed. I am very grateful to Mengfei Cao for sharing my TA burden during my job hunting period. I wish all the best for the friends I made at Tufts.

I would like to thank my mentors at Akamai Technologies, Rosie Jones and Dmitry Pechyony who made my internship days both joyful and fruitful.

I would like to thank all the "family members" at the Tsinghua Alumni Association at Greater Boston (THAA-Boston), in particular Zhizhong Li, Wenjian Lin, Wei Zhang, Xitong Li, Ming Guo, Fei Deng, Zhao Qin, Jiexi Zhang, Zhiting Tian, Yan Zhou, Wei Lin, Fumin Zhou, Zhenke Liu, Sophie Cui, Hai Wang, Yihui Wang, Wujie Huang; because of you all, Boston becomes the "Hub of the universe". I am very lucky to have these wonderful people in my life.

I am happy to thank all my friends in China. In particular, I would like to thank Xianding Wang for being such a great friend. Through my toughest days at college, Min Shui and Cheng Huang supplied me with strength, encouragement and friendship, I would never have gone this far if weren't for you. Thank you!

Special thanks are due to my family, my parents–Xiurong He and Dejun Wang, and my god-parents–Guifen Zhen and Faliang Xu. I appreciate my big sister Ran Yang for always being as supportive as possible. Also, I would like to thank my Fiancée's mom, Yunzhen Tang, who always believes in me and propels me with support and encouragement.

Finally, I would love to thank my Fiancée, Weiwei Gong, for having faith in me, for the tears and laughters, for your patience and tolerance, for your support and encouragement, and mostly important, for your friendship and love!

Contents

List of Figures

List of Tables

Nonparametric Bayesian Mixed-effects Models for Multi-task Learning

Chapter 1

Introduction

This thesis presents several new research developments in *Machine Learning*. Machine learning refers to the science of getting computers to act based on past experience instead of being explicitly programmed (Mitchell, 1997). Machine learning tasks can be broadly divided into three main categories, namely *supervised learning, unsupervised learning* and *reinforcement learning*. In this thesis we will mostly deal with supervised learning on large-scale problems while briefly discussing unsupervised learning.

The framework of supervised learning describes a scenario in which a learner is given a training set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}$ where $\boldsymbol{x}_i$ is normally referred as a *feature vector* and y_i is its corresponding *label*. The goal of a learning algorithm is to estimate a function $f : X \to Y$ such that $f(\boldsymbol{x}^*) \approx y^*$ on any unseen examples $\boldsymbol{x}^*$. Depending on the structure of the output set Y (or labels), one can distinguish between two types of supervised learning problems: *classification* ($Y = \{-1, +1\}$) and *regression* ($Y = \mathbb{R}$).

The mainstream approach is based on *statistical modeling*. That is, one assumes that a statistical model (specified by a set of parameters) generates the observed data; the parameters are estimated using the data; based on these parameters, one can make prediction for unknown quantities of interest. Different research philosophies exist for this process, and in the statistical literature, they are often categorized as the *frequentist approach* and the *Bayesian approach*. These approaches

differ on how they treat the model parameters. The former treats model parameters as fixed unknown quantities while the latter views them being governed by probability distributions (Gelman, 2004). Our methodology in this thesis belongs to the *Bayesian school of statistics*, whose key principle is treating *everything unknown as random variables*.

More specifically, we use *generative models* to describe the process of data generation, where the data includes some observed quantities and some hidden variables. We treat the data "as if" it is sampled using the model we assumed. This process articulates the statistical assumptions that the model makes, and also specifies the *joint probability distribution* (which is proportional to likelihood times the prior distribution) of the hidden and observed random variables. Given the observed data, the primary focus is the problem of *posterior inference*, computing the conditional distribution of the hidden variables given the observed data. Thus, posterior inference is the process of finding the distribution of the hidden structure that likely generated the observed data.

The motivation of this work stems from our collaboration with astrophysicists, who are interested in the problem of classification of stars into meaningful categories. A major effort in astronomy research is devoted to sky surveys, where measurements of the brightness of stars or other celestial objects are taken over a period of time to produce time series, also known as *light curves*. Classification as well as other analyses of stars lead to insights into the nature of our universe. Yet the rate at which data are being collected by these surveys far outpaces current methods to process and classify them.

The data from star surveys is normally represented by *time series* of brightness measurements, based on which they are classified into categories. Stars whose behavior is periodic are especially of interest in such studies. Figure 1-1 shows several examples of such time series generated from the three major types of periodic variable stars: Cepheids, RR Lyraes (RRL), and Eclipsing Binaries (EB).

Clearly, the structure of the time series can be interpreted as a "shape" that is indicative of star type and intuitively, stars from the same class should share

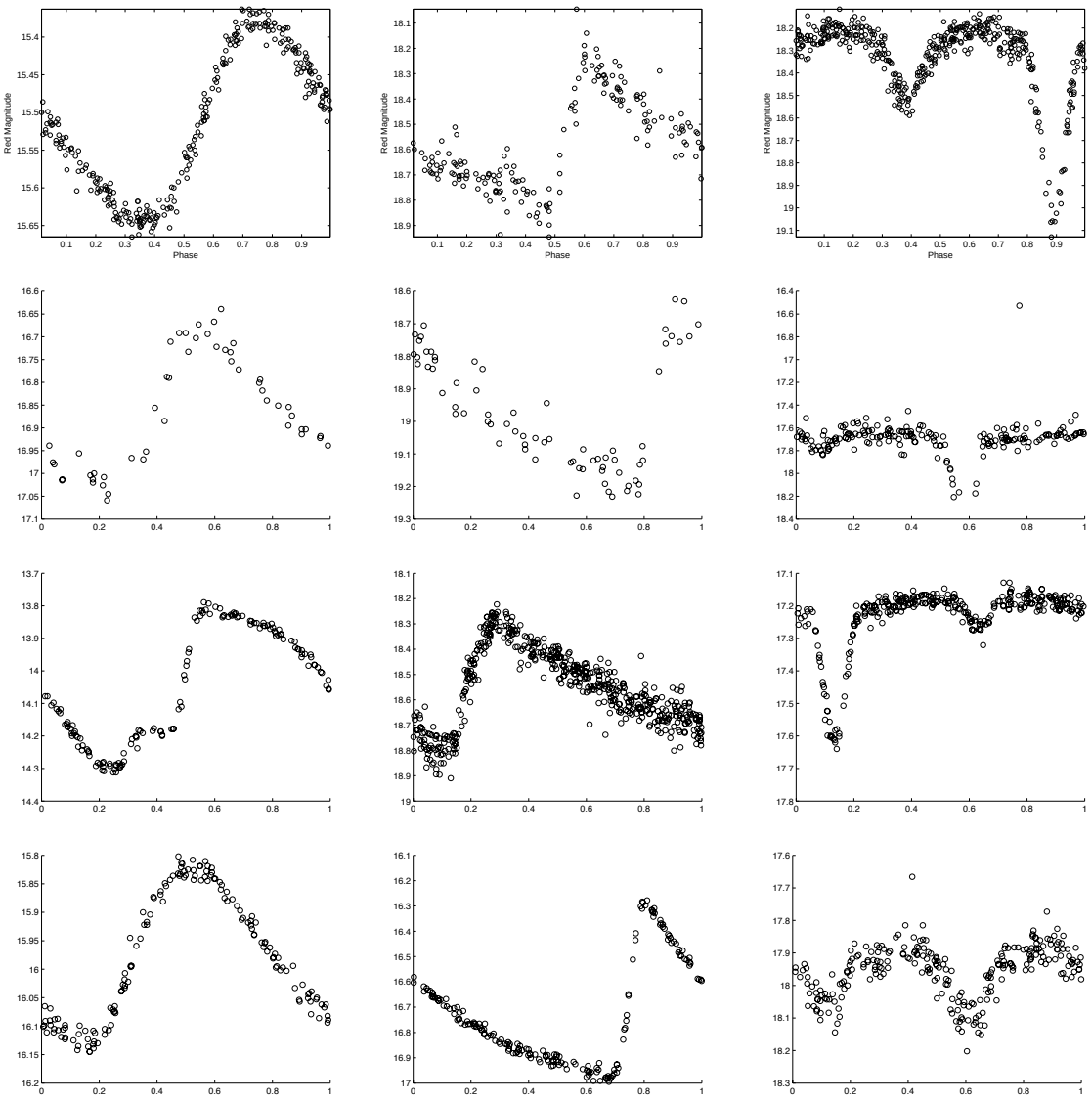

Figure 1-1: Examples of light curves of periodic variable. Left: Cepheid, middle: RR Lyrae, right: Eclipsing Binary.

similar "shape". However, we face a number of challenges, as follows

- Each class has an unknown number of typical "shapes";
- Light curves are not sampled synchronously;
- Each light curve has a typical "shape" associated with the class it belongs to and an individual variation perhaps due to its own physical structure;
- Some light curves have a small number of samples, preventing us from get-

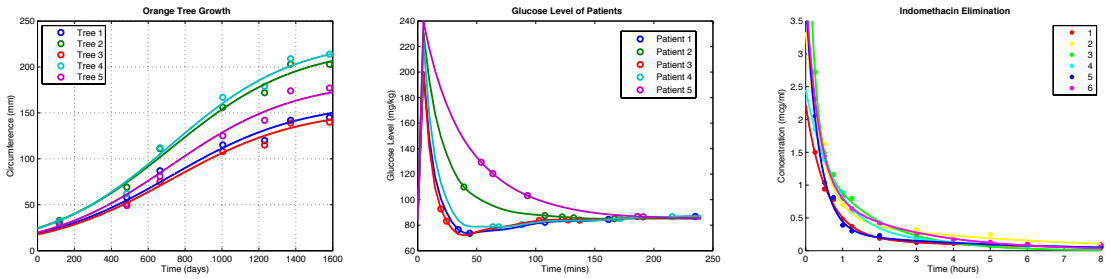

Figure 1-2: Examples that exhibit both group effects and individual effects. Left: Orange Tree Growth (Pinheiro and Bates, 2000, Chap. 10), middle: Glucose Level of different patients after insulin injection (Vicini and Cobelli, 2001; Denti et al., 2010), right: Concentration of drug indomethacin in the bloodstream of six subjects (Courtesy of MATLAB).

ting a clean picture of their "shapes";

- As the light curves are periodic, they can appear in arbitrary phase.

Leaving the phase shift aside for a moment, as shown in Figure 1-2, these properties appear frequently in a wide range of real world applications. For example, in the medical domain, Figure 1-2 (center) shows the glucose level of five different patients after injecting insulin. The corresponding machine learning task is to predict individual patient's glucose level at certain times. One can easily see that the glucose curve of every patient has a group specific shape (perhaps diabetic or not) plus some individual fluctuation (perhaps reflecting the patient's own health condition). Similar observations can be made on data for orange tree growth (Figure 1-2, Left) and concentration of drug indomethacin (Figure 1-2, Right). We will revisit the glucose data set in **Chapter 3**.

In this thesis, we address all the challenges by proposing a family of novel Nonparametric Beyesian models, namely, *Grouped Mixed-effects GP models*. Our models provide solutions as shown in the following table where the terminology and models are given more precisely in **Chapter 2**.

CHALLENGES	SOLUTIONS
multiple typical "shapes"	**mixture model**
non-synchronously sampled data	**regression based model**
individual fluctuation	**mixed-effects model**
sparse samples per times series	**multi-task learning**
phase shift	**specialized phase-shifted regression model**

At a high level, the intuition is as follows: we assume that each class/category has multiple typical "shapes" (we call these "centers" by analogy to clustering models) and each time series that belongs to this class is a sum of one of the typical "shapes" and a specific random fluctuation. This idea, known as *Mixed-effects model*, was developed to handle clustered data and has been a topic of increasing interest in statistics for the past several decades (Demidenko, 2005). In these models the regression function is assumed to be a function of a fixed (population) effect, an instance-specific random effect, and an error term. Observations within the same group share a common effect and are therefore statistically dependent. Linear models are commonly used in mixed-effects modeling and normally the data points are synchronously sampled, for example, each patient is measured at exactly the same time points. Our models make three major extensions: 1) we allow multiple fixed-effects and let the model choose the appropriate number of centers based on the observed data; 2) we learn the grouping rather than assuming that it is known; 3) we model both the typical "shape" and the random "shape" via a nonparametric Bayesian approach, namely, Gaussian Processes (GP). In this way, we allow non-synchronously sampled data.

Nonparametric models have received increasing attention in machine learning in the past two decades. While parametric models require a finite set of parameters capturing all the information contained in the data, nonparametric models make no such assumption and allow the size of models to grow with data size. One example is the *Support Vector Machine (SVM)* or more broadly *kernel machines* where the decision function is a linear combination of the kernel function at every

point in training set, and thus has a complexity that grows with more observations. Nonparametric methods hold significant advantages in wide range of real world applications (Joachims, 2002; Lafferty et al., 2004; Kocijan et al., 2004; Joachims, 2005; Chu et al., 2005; Murillo-Fuentes et al., 2006).

Nonparametric Bayesian methods put a prior distribution over the unknown (infinite set of) parameters. Therefore the prior distributions become random functions, or more precisely, *stochastic processes*. Of them, the most elegant is the Gaussian process (GP) model, which generalizes the multivariate normal distributions. In recent years, GPs have gained popularity due to their ability to produce probabilistic predictions and their explicit way of modeling correlation.

Chapter 2 gives an overview of Gaussian process modeling, with a focus on regression. We provide background on multi-task learning and introduce the *Mixed-effects GP models* and *Grouped Mixed-effects GP models* (GMT).

In **Chapter 3**, we focus on the original problem that motivated our thesis work, astrophysics time series classification. Since we are interested in periodic stars, there is one extra challenge: the time series are not phase aligned, meaning that light curves in the same category share a similar shape but with some unknown shift. Therefore, we modify our model to incorporate the phase shift explicitly (yielding the Shift-invariant GMT model) and develop an efficient inference algorithm based on the EM algorithm. As a special case we obtain the Gaussian mixture model for phased-shifted periodic time series. As in other mixture models, setting the correct number of centers is an important yet challenging problem. A common solution is to use a model selection metric which usually includes two terms to compare models of different orders. The first term measures how well the model fits the data. The second term, a complexity penalty, favors simpler models (i.e., ones with fewer centers). In Chapter 3, we use another important tool from the Nonparametric Bayesian literature, the *Dirichlet Processes*. The idea is as follows: rather than comparing models that vary in complexity, the DP approach fits a single model that can adapt its complexity to the data. Thus, by using a *Dirichlet Process* prior over the mixture proportion, the model estimates how many clusters

are needed to model the observed data and allows future data to exhibit previously unseen clusters. This leads to an infinite mixture model (the DP-GMT model) that is capable of automatic model selection. This Chapter is based on (Wang et al., 2010, 2011).

However, one of the main difficulties with this model, is the computational cost. The time complexity of GP training scales cubically with the number of training data points N, and prediction scales quadratically in N. GPs in their standard form have therefore been limited in their applicability to data sets of only several thousand data points. In our case, while the number of samples per task is small, the total sample size can be large, and the typical cubic complexity of GP inference can be prohibitively large. Previous work showed that some improvement can be obtained when all the input tasks share the same sampling points, or when different tasks share many of the input points. However, if the number of distinct sampling points is large the complexity remains high. To address this issue, in **Chapter 4** we propose a sparse model with variational model selection that greatly speeds up learning without degrading performance. The key idea is to compress the information of all tasks into an optimal set of pseudo samples for each mean effect. Our approach is particularly useful when individual tasks have a small number of samples, different tasks do not share sampling points, and there is a large number of tasks. This Chapter is based on (Wang and Khardon, 2012b,a).

Another challenge in classification of astrophysics time series is to determine whether a light curve is a variable star and to further estimate its period. State-of-the-art performance requires a human in the loop to verify that the star is in fact periodic, and that the period-finder has returned the true period. In **Chapter 5**, we cast this problem as a model selection problem under the GP regression framework. We develop a novel algorithm for parameter optimization for GP which is useful when the likelihood function is very sensitive to the parameters with numerous local minima, as in the case of period estimation. In particular, we propose and evaluate: an approximation using a two level grid search, an approximation using limited cyclic optimization, a method using sub-sampling and averaging,

and a method using low-rank Cholesky approximations. Experimental results validate our approach showing significant improvement over existing methods. This Chapter is based on (Wang et al., 2012).

Finally, in **Chapter 6**, we conclude and outline directions for future work. In summary, our contributions are

1. We propose the *Grouped Mixed-effects GP* models, a family of novel Nonparametric Bayesian models for multi-task learning, and develop computationally efficient inference algorithms for these models. Our models are well suited for *time series prediction, classification and clustering*.

2. We extend the aforementioned *Grouped Mixed-effects GP* model to handle phase shift, yielding the *Shift-invariant Grouped Mixed-effects GP* (GMT). We also propose its *infinite* version, DP-GMT that can choose model order automatically. For both models, we develop inference algorithms, namely, an EM algorithm for the GMT model and a Variational EM algorithm for DP-GMT optimizing the maximum a posteriori (MAP) estimates for the parameters of the models.

3. To make the *Grouped Mixed-effects GP* "big data friendly", we develop a *sparse variational learning algorithm* that reduces the computational complexity significantly. As a special case, we recover the sparse algorithm for singe-task GP regression proposed in Titsias (2009).

4. We develop a new algorithm for period estimation using the GP regression model by developing a novel method for model selection for GP. Our algorithm combines gradient optimization with grid search and incorporates several mechanisms to improve the complexity over the naive approach. In addition, we also propose a novel approach for using domain knowledge, in the form of a probabilistic generative model, and incorporate it into the period estimation algorithm.

Chapter 2

Gaussian Processes and Multi-task Learning

Our work makes extensive use of kernel based learning with Gaussian processes (GP) and properties of multivariate normal distributions. For an in depth introduction to this area, the reader is encouraged to consult (Rasmussen and Williams, 2006). We also assume familiarity with general principles and techniques in *Pattern Recognition*, for example, as covered by (Bishop, 2006). In this chapter we review some of the basic results and algorithms for learning with GP and then define the multi-task learning and mixed-effects model studied in this thesis. More specific tools and techniques are discussed in the context of their usage in individual chapters, in particular, Expectation Maximization in Section 3.2.1, Dirichlet processes and Variational EM in Section 3.3.

Notation

Throughout the thesis, scalars are denoted using italics, as in $x, y \in \mathbb{R}$; vectors (column vectors by default) use bold typeface, as in $\boldsymbol{x}, \boldsymbol{y}$, and x_i denotes the ith entry of $\boldsymbol{x}$. For a vector $\boldsymbol{x}$ and real valued function $f : \mathbb{R} \to \mathbb{R}$, we extend the notation for f to vectors so that $f(\boldsymbol{x}) = [f(x_1), \cdots, f(x_n)]^{\mathrm{T}}$ where the superscript T stands for transposition (and the result is a column vector). $\mathcal{K}(\cdot, \cdot)$ denotes a

kernel function associated with some reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ and its norm is denoted as $\|\cdot\|_{\mathcal{H}}$, i.e. $\|f\|_{\mathcal{H}}^2 = \mathcal{K}(f,f)$. To keep the notation simple, $\sum_{j=1}^{M}$ or $\prod_{j=1}^{M}$ is substituted by $\sum_j$ or $\prod_j$ where the index j is not confusing.

Gaussian distributions are a convenient modeling tool because their analytical properties make a wide range of mathematics and engineering problems tractable. These nice properties hold even when we increase the dimensionality from 1 to infinity. Let us first introduce the multivariate Normal distribution. An n-dimensional random vector $\boldsymbol{x}$ that follows a multivariate normal distribution of mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, denoted as $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, has a probability density (pdf)

$$(2\pi)^{-n/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}.$$

The following formulas for multivariate normal distributions are used repeatedly throughout this thesis. Let

$$\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^{\mathrm{T}} & \mathbf{C}\end{bmatrix}\right),$$

then the marginal distribution of $\boldsymbol{x}$ and $\boldsymbol{y}$ are

$$\begin{aligned}\boldsymbol{x} &\sim \mathcal{N}(\boldsymbol{\mu}_x, \mathbf{A})\\ \boldsymbol{y} &\sim \mathcal{N}(\boldsymbol{\mu}_y, \mathbf{C}).\end{aligned}$$

The conditional distribution is

$$\boldsymbol{x}|\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}_x + \mathbf{B}\mathbf{C}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_y), \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^{\mathrm{T}})$$

Let $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{y}|\boldsymbol{x} \sim \mathcal{N}(\mathbf{A}\boldsymbol{x}+\boldsymbol{b}, \boldsymbol{\Lambda})$, then we have the marginal distribution of $\boldsymbol{y}$,

$$\boldsymbol{y} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu}+\boldsymbol{b}, \boldsymbol{\Lambda} + \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^{\mathrm{T}}).$$

2.1 Gaussian Processes for Machine Learning

2.1.1 Learning and Inference

A Gaussian process (GP) is a functional extension for Multivariate Gaussian distributions. In the Bayesian literature, it has been widely used in statistical models by substituting a parametric latent function with a stochastic process with a Gaussian prior (Rasmussen and Williams, 2006). To explain this approach, we start with the following regression model,

$$y = f_{\boldsymbol{w}}(\boldsymbol{x}) + \epsilon, \tag{2.1}$$

where $f_{\boldsymbol{w}}(\boldsymbol{x})$ is the regression function with parameter $\boldsymbol{w}$ and ϵ is independent identically distributed (iid) Gaussian noise with variance σ^2. For example, in linear regression $f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ and therefore $y \sim \mathcal{N}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}, \sigma^2)$. Given the data $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}, i = 1, \cdots, N$, one wishes to infer $\boldsymbol{w}$ and the basic approach is to maximize the likelihood

$$\mathcal{L}(\boldsymbol{w}|\mathcal{D}) = \Pr(\mathcal{D}|\boldsymbol{w}) = \prod_{i=1}^{N} \Pr(y_i|\boldsymbol{x}_i, \boldsymbol{w}).$$

In Bayesian statistics, the parameter $\boldsymbol{w}$ is assumed to have a prior probability $\Pr(\boldsymbol{w})$ which encodes the prior belief on the parameter. The inference task becomes calculating the posterior distribution over $\boldsymbol{w}$, which, using the Bayes formula, is given as

$$\Pr(\boldsymbol{w}|\mathcal{D}) \propto \Pr(\mathcal{D}|\boldsymbol{w}) \Pr(\boldsymbol{w}). \tag{2.2}$$

The predictive distribution for a new observation $\boldsymbol{x}^*$ is given by

$$\Pr(f(\boldsymbol{x}^*)|\mathcal{D}) = \int \Pr(f(\boldsymbol{x}^*)|\boldsymbol{w}) \Pr(\boldsymbol{w}|\mathcal{D}) d\boldsymbol{w}. \tag{2.3}$$

Returning to linear regression, the common model assumes that the prior for $\boldsymbol{w}$ is a zero-mean multivariate Gaussian distribution, and the posterior turns out to be multivariate Gaussian as well. Generally, calculating the posterior distribution is difficult in Bayesian statistics. However, in this case, as well as in GP models defined below, we often have simple algorithms for inference or calculation of

desired quantities because of the nice properties of multivariate Gaussian distributions and corresponding facts from linear algebra.

This approach can be made more general using a nonparametric Bayesian model. In this case we replace the parametric latent function f_w by a stochastic process f where f's prior is given by a Gaussian process (GP). A GP is specified by a mean function $m(\cdot)$ and covariance function $\mathcal{K}(\cdot,\cdot)$. This allows us to specify a prior over functions f such that the distribution induced by the GP over any finite sample is normally distributed. The particular choice of covariance function determines the properties of sample functions drawn from the GP prior (e.g. smoothness, length scales, amplitude etc).

More precisely, the GP regression model with zero mean and covariance function $\mathcal{K}(\cdot,\cdot)$ is as follows. Given sample points $[\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n]^{\mathrm{T}}$ let $\mathbf{K} = (\mathcal{K}(\boldsymbol{x}_i,\boldsymbol{x}_j))_{i,j}$. The induced distribution on the values of the function at the sampling points is

$$\mathbf{f} \triangleq [f(\boldsymbol{x}_1),\cdots,f(\boldsymbol{x}_n)]^{\mathrm{T}} \sim \mathcal{N}(\mathbf{0},\mathbf{K}), \tag{2.4}$$

where $\mathcal{N}$ denotes the multivariate normal distribution. Now assuming that y_i is generated from $f(\boldsymbol{x}_i)$ using iid noise as in (2.1) and denoting $\boldsymbol{y} = [y_1,\ldots,y_n]^{\mathrm{T}}$ we get that $\boldsymbol{y} \sim \mathcal{N}(\mathbf{0},\mathbf{K}+\sigma^2\mathbb{I})$ and the joint distribution is given by

$$\begin{bmatrix}\mathbf{f}\\ \boldsymbol{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{K} & \mathbf{K}\\ \mathbf{K} & \mathbf{K}+\sigma^2\mathbb{I}\end{bmatrix}\right). \tag{2.5}$$

Using properties of multivariate Gaussians we can see that the posterior distribution $\mathbf{f}|\boldsymbol{y}$ is given by

$$\Pr(\mathbf{f}|\boldsymbol{y}) = \mathcal{N}(\mathbf{K}\,(\sigma^2\mathbb{I}+\mathbf{K})^{-1}\,\boldsymbol{y}\,,\,\mathbf{K}(\sigma^2\mathbb{I}+\mathbf{K})^{-1}\mathbf{K}). \tag{2.6}$$

Similarly, given the observed data $\mathcal{D}$, the predictive distribution for some test point

$\boldsymbol{x}_*$ distinct from the training examples is given by

$$\begin{aligned}\Pr(f(\boldsymbol{x}_*)|\boldsymbol{x}_*, \mathcal{D}) &= \int \Pr(f(\boldsymbol{x}_*)|\boldsymbol{x}_*, f) \Pr(f|\mathcal{D}) df \\ &= \mathcal{N}\Big(\mathbf{k}(\boldsymbol{x}_*)^{\mathrm{T}}(\sigma^2\mathbb{I} + \mathbf{K})^{-1}\boldsymbol{y}, \\ &\qquad \mathcal{K}(\boldsymbol{x}_*, \boldsymbol{x}_*) - \mathbf{k}(\boldsymbol{x}_*)^{\mathrm{T}}(\sigma^2\mathbb{I} + \mathbf{K})^{-1}\mathbf{k}(\boldsymbol{x}_*)\Big)\end{aligned} \tag{2.7}$$

where $\mathbf{k}(\boldsymbol{x}_*) = [\mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_*), \cdots, \mathcal{K}(\boldsymbol{x}_N, \boldsymbol{x}_*)]^{\mathrm{T}}$. Therefore, although the model is based on a distribution over functions, inference and prediction can be done directly using finite samples. Figure 2-1 illustrates GP regression, by showing how a finite sample induces a posterior over functions and their predicted values for new sample points.

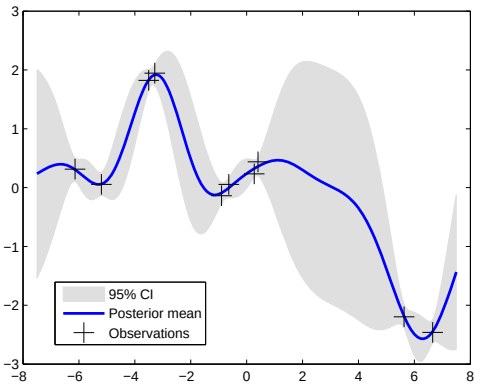

Figure 2-1: Illustration of prediction with GP regression. The data points $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}$ are given by the crosses. The shaded area represents the pointwise 95% confidence region of the predictive distribution. As can be seen from (2.7), GP regression can be seen to perform a variant of kernel regression where $f(\boldsymbol{x}_*)$ is a weighted average of all the measurements $\boldsymbol{y}$. While the values of the weights are obscured because of the inverse of the covariance matrix in that expression, one can view this roughly by an analogy to nearest neighbor regression where the mean of $f(\boldsymbol{x}*)$ is affected more by the measurements whose sampling points are "close" to $\boldsymbol{x}^*$ and the variance of $f(\boldsymbol{x}^*)$ is small if $\boldsymbol{x}^*$ is surrounded by measurements. A deeper discussion of the equivalent kernel is given in (Rasmussen and Williams, 2006).

Readers with background in kernel-based machine learning may find the form

of the posterior and predictive distribution very familiar. Indeed, the posterior mean of the GPR is exactly the solution to the following least squares problem,

$$\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - y)^2 + \sigma^2\|f\|_{\mathcal{H}},$$

where $\|f\|_{\mathcal{H}}$ denotes the RKHS norm of the function f. One can relate a Gaussian process f to a RKHS $\mathcal{H}$ with kernel $\mathcal{K}$ such that

$$\text{cov}[f(\boldsymbol{x}), f(\boldsymbol{y})] = \mathcal{K}(\boldsymbol{x}, \boldsymbol{y}) \quad \forall \boldsymbol{x}, \boldsymbol{y} \in X. \tag{2.8}$$

In this way, we can express a prior on functions f using a zero mean Gaussian process (Lu et al., 2008).

$$f \sim \exp\left\{-\frac{1}{2}\|f\|_{\mathcal{H}}^2\right\}. \tag{2.9}$$

Though tempting as it sounds, in general, a Gaussian process cannot be thought of as a distribution on the RKHS, because with probability 1, one can find a Gaussian process such that its sample path does not belong to the RKHS. However, the equivalence holds between the RKHS and the expectation of a Gaussian process conditioned on a finite number of observations. For more details on the relationship between RKHS and Gaussian processes we refer interested readers to Seeger (2004).

2.1.2 Model Selection

From the previous section, we know how to perform GPR inference when the covariance function is fully specified. However, in real applications, it is not easy to determine the parameters of the covariance function. Thus, we need a criterion to help us choose the free parameters (hyper-parameters) in the covariance function.

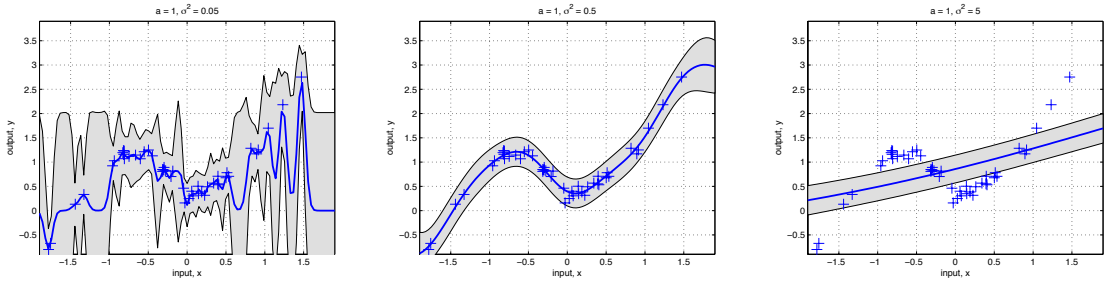

Figure 2-2: Estimated Curves using different values of hyper-parameters. The blue plus are observations and the solid curves are estimated functions. The shaded areas are 0.95 confidence region. Data is generated from a zero mean GP with RBF kernel with $a = 1, \sigma^2 = 0.5$. Three pictures show the learned curves with $a = 1$ but using different σ^2's. Left: $\sigma^2 = 0.05$; Center: $\sigma^2 = 0.5$; Right: $\sigma^2 = 5$.

For example, consider the *Radial Basis Function* (RBF kernel), i.e.,

$$\mathcal{K}(x, y) = a \exp\left\{-\frac{\|x - y\|^2}{2\sigma^2}\right\}, \tag{2.10}$$

The two hyper-parameters a and σ^2 govern properties of sample functions where a controls the typical amplitude and σ^2 controls the typical length scale of variation. We refer to any hyper-parameters of a covariance function collectively as the vector $\boldsymbol{\theta}$. Clearly, we can see that function variables close in input space are highly correlated, whereas function variables far apart relative to the length scale σ^2 are uncorrelated.

Figure 2-2 shows that severe over/under fitting could happen when one chooses the wrong hyper-parameters. Therefore, selecting an appropriate model is of vital importance in real world applications. We next review two most popular approaches. A more detailed treatment can be found in (Rasmussen and Williams, 2006).

Marginal Likelihood

The standard approach, known as *empirical Bayes*, is to identify the hyper-parameters that maximize the marginal likelihood. More precisely, we try to find an appropri-

ate model $\mathcal{M}^*$ such that

$$\mathcal{M}^* = \underset{\mathcal{M}}{\operatorname{argmax}} \left[\log \left[\Pr(\boldsymbol{y}|\boldsymbol{x}; \mathcal{M})\right]\right] \tag{2.11}$$

where the logarithm of the marginal likelihood is given by

$$\begin{aligned} \log \Pr(\boldsymbol{y}|\boldsymbol{x}; \mathcal{M}) &= \log \left(\int \Pr(\boldsymbol{y}|f, \boldsymbol{x}; \mathcal{M}) \Pr(f|\boldsymbol{x}; \mathcal{M}) \, df \right) \\ &= -\frac{1}{2} \boldsymbol{y}^{\mathrm{T}} (\mathbf{K} + \sigma^2 \mathbb{I})^{-1} \boldsymbol{y} \\ &\quad - \frac{1}{2} \log |\mathbf{K} + \sigma^2 \mathbb{I}|^{-1} - \frac{n}{2} \log 2\pi \end{aligned} \tag{2.12}$$

and (2.12) holds because $\boldsymbol{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2 \mathbb{I})$ (Rasmussen and Williams, 2006). We can see that the hidden function f is marginalized out, hence the name "marginal likelihood". Typically, one can optimize the marginal likelihood by calculating the partial derivative of the marginal likelihood with respect to (w.r.t.) the hyper-parameters and optimizing the hyper-parameters using gradient based search (Rasmussen and Williams, 2006). The partial derivative of (2.12) w.r.t. the parameter θ_j is (Rasmussen and Williams, 2006)

$$\frac{\partial}{\partial \theta_j} \log \Pr(\boldsymbol{y}|\boldsymbol{x}; \mathcal{M}) = \operatorname{Tr} \left(\left(\boldsymbol{\alpha} \boldsymbol{\alpha}^{\mathrm{T}} - \mathbf{K}_\sigma^{-1} \right) \frac{\partial \mathbf{K}_\sigma}{\partial \theta_j} \right) \tag{2.13}$$

where $\mathbf{K}_\sigma = \mathbf{K} + \sigma^2 \mathbb{I}$ and $\boldsymbol{\alpha} = \mathbf{K}_\sigma^{-1} \boldsymbol{y}$.

In this thesis, we mostly use this method with gradient based optimization. However, in Chapter 5 where we study period estimation, the marginal likelihood becomes very sensitive to the hyper-parameters and thus gradient alone fail to work. We therefore develop a more elaborate method in Chapter 5.

Cross-Validation

An alternative approach (Rasmussen and Williams, 2006) picks hyperparameter $\mathcal{M}$ by minimizing the empirical loss on a hold out set. This is typically done with a leave-one-out (LOO-CV) formulation, which uses a single observation from the

original sample as the validation data, and the remaining observations as the training data. The process is repeated such that each observation in the sample is used once as the validation data. To be precise, we choose the hyper-parameter $\mathcal{M}^*$ such that

$$\mathcal{M}^* = \underset{\mathcal{M}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \hat{f}_{-i}(x_i))^2 \tag{2.14}$$

where $\hat{f}_{-i}$ is defined as the posterior mean given the data $\{\boldsymbol{x}_{-i}, \boldsymbol{y}_{-i}\}$ in which the subscript $-i$ means all but the ith sample, that is,

$$\hat{f}_{-i}(x) = \mathcal{K}(\boldsymbol{x}_{-i}, x)^T \left(\mathbf{K}_{-i} + \sigma^2 \mathbb{I}\right)^{-1} \boldsymbol{y}_{-i}. \tag{2.15}$$

It can be shown that this computation can be simplified (Rasmussen and Williams, 2006) using the fact that

$$y_i - \hat{f}_{-i}(x_i) = \frac{\left[(\mathbf{K} + \sigma^2 \mathbb{I})^{-1} \boldsymbol{y}\right]_i}{\left[(\mathbf{K} + \sigma^2 \mathbb{I})^{-1}\right]_{ii}} \tag{2.16}$$

where $[\cdot]_i$ is the ith entry of the vector and $[\cdot]_{ii}$ denotes the (i,i)th entry of the matrix. We will compare this approach to the marginal likelihood approach in Chapter 5. In Chapter 3 and 4, we use gradient based optimization of marginal likelihood to select hyper-parameters.

2.2 Multi-task Learning

In real world problems, there are situations when multiple learning tasks are need while the training set for each task is quite small. For example, in pharmacological studies, we may be attempting to predict the blood concentration of a medicine at different times across multiple patients. Finding the best-fit function for a single patient based only on his measurements makes the learning difficult. Instead, if we can get strength from measurements across all the patients, it is more likely to estimate a function that better generalizes to the population at large. Multi-task

learning arises in favor of learning multiple correlated task simultaneously. Over the last decade, this topic has been the focus of much interest in the machine learning literature. Several approaches have been applied to a wide range of domains, such as medical diagnosis, (Bi et al., 2008), recommendation systems (Dinuzzo et al., 2008), HIV Therapy Screening (Bickel et al., 2008).

Consider the standard supervised learning problem, given a training set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}, i = 1, \cdots, N$, where $\boldsymbol{x}_i \in X \subset \mathbb{R}^d$, single-task learning focuses on finding a function $f : X \to \mathbb{R}$, where $X \subset \mathbb{R}^d$, which best fits and generalizes the observed data. A fundamental limitation is the cost incurred by the preparation of the large training samples required for good generalization. A potential remedy is offered by multi-task learning: in many cases, while individual sample sizes are rather small, there are samples to represent a large number of learning tasks, which share some constraining or generative property. More precisely, suppose we have a set of M (related) tasks, with jth task generated data $\mathcal{D}^j = (\boldsymbol{x}_i^j, y_i^j), i = 1, 2, \cdots, N_j$. We are interested in finding M functions for each task. If one learns each task separately, it boils down to single-task learning. However, when there are relations between the tasks to learn, it can be advantageous to learn all tasks simultaneously instead of following the more traditional approach of learning each task independently of the others. Thus, exploring a common property using the entire collection of training samples, it should allow better estimation of the individual tasks despite their small individual sample sizes. Indeed, the information provided by data for a specific task may serve as a domain- specific inductive bias for the others.

In the past decade, starting with the seminal work of Caruana (1997), multi-task learning, also under names such as *transfer learning,* and *learning to learn*, has been extensively studied in the Machine Learning community (Bakker and Heskes, 2003; Micchelli and Pontil, 2005; Xue et al., 2007a; Yu et al., 2005; Schwaighofer et al., 2005; Pillonetto et al., 2010; Pan and Yang, 2010). There are two main formulations of multi-task learning that are closely related to the work in this thesis, namely in terms of the regularization framework and in the use of a Bayesian ap-

proach.

2.2.1 Regularization Formation

In particular, in the regularization framework, we wish to solve the following problem (Micchelli and Pontil, 2005)

$$\underset{f^1,\cdots,f^M\in\mathcal{H}}{\operatorname{argmin}} \left\{ \frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{N_j}(y_i^j - f^j(\boldsymbol{x}_i^j))^2 + \lambda \mathrm{PEN}(f^1, f^2, \cdots, f^M) \right\} \tag{2.17}$$

where the penalty term, applying jointly to all the tasks, encodes our prior information on how smooth the functions are, as well as how these tasks are correlated with each other. For example, setting the penalty term to $\sum_j \|f^j\|_{\mathcal{H}}$ implies that there is no correlation among the tasks. It further decomposes the optimization functional to M separate single-task learning problems. On the other hand, with a shared penalty, the joint regularization can lead to improved performance.

Moreover, we can use a norm in RKHS with a *multi-task kernel* to incorporate the penalty term (Micchelli and Pontil, 2005). Formally, consider a vector-valued function $f : X \to \mathbb{R}^M$ defined as $f \triangleq [f^1, f^2, \cdots, f^M]^{\mathrm{T}}$. Then Equation (2.17) can be written as

$$\underset{f\in\mathcal{H}}{\operatorname{argmin}} \left\{ \frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{N_j}(y_i^j - f^j(\boldsymbol{x}_i^j))^2 + \lambda \|f\|_{\mathcal{H}}^2 \right\} \tag{2.18}$$

where $\|\cdot\|_{\mathcal{H}}$ is the norm in RKHS with the multi-task kernel $\mathcal{Q} : (\boldsymbol{\Lambda}, X) \times (\boldsymbol{\Lambda}, X) \to \mathbb{R}$, where $\boldsymbol{\Lambda} = \{1, 2, \cdots, M\}$. As shown by Evgeniou et al. (2006), the *representer theorem* shows that the solution to (2.18)

$$f^{\ell}(\cdot) = \sum_{j=1}^{M}\sum_{i=1}^{N_j} c_i^j \mathcal{Q}((\cdot, \ell), (\boldsymbol{x}_i^j, j)) \tag{2.19}$$

with norm

$$\|f\|_{\mathcal{Q}}^2 = \sum_{\ell,k}\sum_{i=1}^{n_\ell}\sum_{j=1}^{n_k} c_i^\ell c_j^k \mathcal{Q}((\boldsymbol{x}_i^\ell, \ell), (\boldsymbol{x}_j^k, k)).$$

Let $\mathbf{C} = [c_1^1, c_2^1, \cdots, c_{n_M}^M]^{\mathrm{T}}, \mathbf{Y} = [y_1^1, y_2^1, \cdots, y_{n_M}^M]^{\mathrm{T}} \in \mathbb{R}^{\sum_j N_j}$ and $\mathbf{X} = [\boldsymbol{x}_1^1, \boldsymbol{x}_2^1, \cdots, \boldsymbol{x}_{n_M}^M]$. Using this formation, the coefficients $\{c_i^j\}$ are given by the following linear system

$$(\mathbf{Q} + \lambda \mathbb{I})\mathbf{C} = \mathbf{Y} \tag{2.20}$$

where $\mathbf{Q} \in \mathbb{R}^{\sum_j N_j \times \sum_j N_j}$ is the kernel matrix formed by $\mathbf{X}$.

2.2.2 GP Model for Multi-task Learning

On the other hand, several approaches formalizing multi-task learning exist within Bayesian statistics. Considering hierarchical Bayesian models (Xue et al., 2007b; Gelman, 2004), one can view the parameter sharing of the prior among tasks as a form of multi-task learning where evidence from all tasks is used to infer the parameters. Over the past few years, Bayesian models for multi-task learning were formalized using Gaussian processes (Yu et al., 2005; Schwaighofer et al., 2005; Pillonetto et al., 2010). One particularly interesting Multi-task GP model is proposed by Bonilla et al. (2008) that learns a shared covariance matrix on features and a covariance matrix for tasks that explicitly models the dependency between tasks, i.e. $\mathbf{Cov}(f^i(s), f^j(t)) = \mathbf{C}(i,j) \times \mathcal{K}(s,t)$.

In this thesis, we concentrate on the so-called *nonparametric Bayesian mixed-effects model* (Lu et al., 2008; Pillonetto et al., 2010). In this mixed-effects model, information is shared among tasks by having each task f^j (associated with the dataset $\mathcal{D}^j$) combine a common (fixed effect) portion and a task specific portion, each of which is generated by an independent Gaussian process.

Assumption 1 (Mixed-effects GP) *For each j and $\boldsymbol{x} \in X$,*

$$f^j(\boldsymbol{x}) = \bar{f}(\boldsymbol{x}) + \tilde{f}^j(\boldsymbol{x}), \quad j = 1, \cdots, M. \tag{2.21}$$

where $\bar{f}$ and $\tilde{f}^j$ are zero-mean Gaussian processes. In addition, $\bar{f}$ and $\tilde{f}^j$ are assumed to be mutually independent.

This assumes that the fixed-effect (mean function) $\bar{f}(\boldsymbol{x})$ is sufficient to capture the

behavior of the data, an assumption that is problematic for distributions with several modes. To address this, we introduce a mixture model allowing for multiple modes (just like standard Gaussian mixture model (GMM)), but maintaining the formulation using Gaussian processes. This amounts to adding a group effect structure and yields the central object of our study in this thesis:

Assumption 2 (Grouped Mixed-effects GP (GMT)) *For each j and $\boldsymbol{x} \in X$,*

$$f^j(\boldsymbol{x}) = \bar{f}_{z_j}(\boldsymbol{x}) + \tilde{f}^j(\boldsymbol{x}), \quad j = 1, \cdots, M \tag{2.22}$$

where $\{\bar{f}_s\}, s = 1, \cdots, k$ and $\tilde{f}^j$ are zero-mean Gaussian processes and $z_j \in \{1, \cdots, k\}$. In addition, $\{\bar{f}_s\}$ and $\tilde{f}^j$ are assumed to be mutually independent.

We can connect this model back to the regularization framework. With the grouped-effect model and groups predefined, one can define a kernel that relates (with non zero similarity) only points from the same example or points for different examples but the same center as follows

$$\mathcal{Q}((\boldsymbol{x}, i), (\boldsymbol{x}', j)) = \delta_{z_i, z_j} \overline{\mathcal{K}}_{z_i}(\boldsymbol{x}, \boldsymbol{x}') + \delta_{i,j} \widetilde{\mathcal{K}}_i(\boldsymbol{x}, \boldsymbol{x}')$$

where

$$\begin{cases} \overline{\mathcal{K}}_{z_i}(\boldsymbol{x}, \boldsymbol{x}') = \operatorname{cov}[\bar{f}_{z_i}(\boldsymbol{x}), \bar{f}_{z_i}(\boldsymbol{x}')], \\ \widetilde{\mathcal{K}}_i(\boldsymbol{x}, \boldsymbol{x}') = \operatorname{cov}[\tilde{f}^j(\boldsymbol{x}), \tilde{f}^j(\boldsymbol{x}')]. \end{cases}$$

We point the connection out for completeness but will not use this formation in the rest of the thesis.

In Chapter 3, we will discuss an extension of Grouped Mixed-effects GP that is capable of handling phase shifted time series data. As will become clear there, the complexity of inference in GMT can be prohibitively high when we have many tasks. We will address the time complexity issue of the proposed model in Chapter 4 by developing a so-called variational sparse solution for the model. Chapter 5 goes back to the problem of inference in single task GP when applied to the period estimation problem.

Our implementation of all the algorithms introduced in this thesis makes extensive use of the GPML package (Rasmussen and Nickisch, 2010) and extends it to implement the required functions.

Chapter 3

Shift-invariant Grouped Mixed-effects GP

In some applications, we face the challenge of learning a periodic function on a single period from samples. Motivated by an application in Astrophysics, we consider the case where the mean effect functions in the same group may differ in their phase. We are interested in classifying stars into different categories based on the time series like the data shown in Figure 3-1. It can be noticed that there are two main characteristics of this data set:

- The time series are not phase aligned, meaning that the light curves in the same category share a similar shape but with some unknown shift.
- The time series are non-synchronously sampled and each light curve has a different number of samples and sampling times.

These characteristics were our motivation for the main object introduced in this chapter, which extends the grouped mixed-effects model (see Assumption 2) such that each task may be an arbitrarily phase-shifted image of the original time series. We call this model *Shift-invariant Grouped Mixed-effects Model* (GMT), which allows us to handle phase shifted time series. In particular, we have the following model:

Assumption 3 (Shift-invariant Grouped mixed-effects Model) *For the j-th task and*

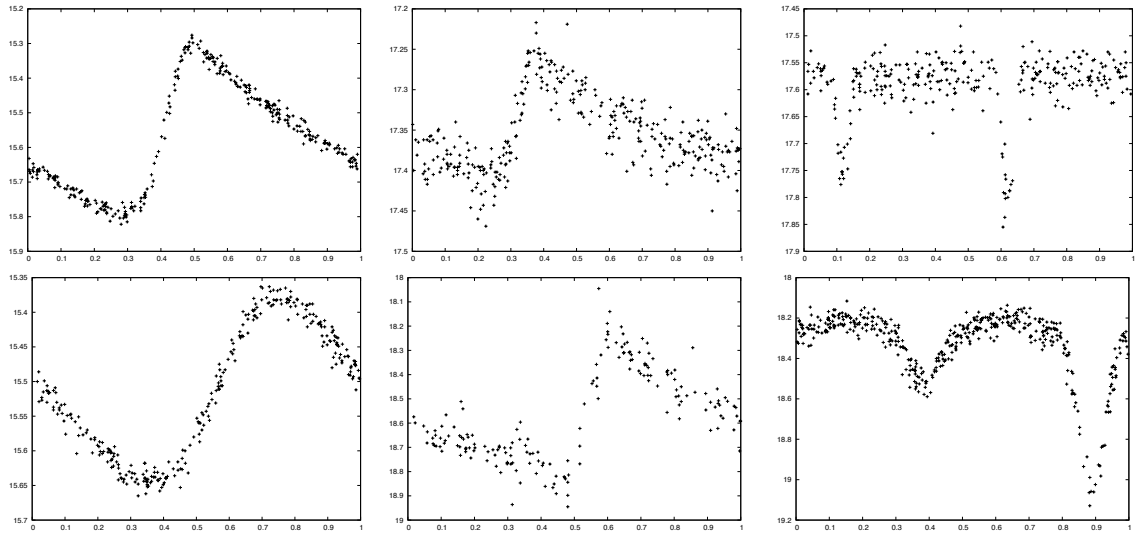

Figure 3-1: Examples of light curves of periodic variable stars folded according to their period to highlight the periodic shape. Left: Cepheid, middle: RR Lyrae, right: Eclipsing Binary.

$x \in [0, T)$,

$$f^j(x) = [\bar{f}_{z_j} * \delta_{t_j}](x) + \tilde{f}^j(x), \quad j = 1, \cdots, M \tag{3.1}$$

where $z_j \in \{1, \cdots, k\}$, $\{\bar{f}_s\}, s = 1, \cdots, k$ *and* $\{\tilde{f}^j\}$ *are zero-mean Gaussian processes,* $*$ *stands for circular convolution and* δ_{t_j} *is the Dirac* δ *function with support at* $t_j \in [0, T)$. *In addition,* $\{\bar{f}_s\}, \tilde{f}^j$ *are assumed to be mutually independent.*

Remark 1 *Given a periodic function* f *with period* T*, its circular convolution with another function* h *is defined as*

$$(f * h)(t) \triangleq \int_{t_0}^{t_0+T} f(t-\tau)h(\tau)d\tau$$

where t_0 *is arbitrary in* $\mathbb{R}$ *and* $f * h$ *is also a periodic function with period* T*. Using the definition we see that,*

$$f * \delta_{t_j}(t) = f(t - t_j),$$

and thus $*$ *performs a right shift of* f *or in other words performs a phase shift of* t_j *on* f*.*

Alternatively, our model can be viewed as a probabilistic extension of the Phased K-means algorithm of Rebbapragada et al. (2009) that performs clustering for phase-

shifted time series data, and as a non-parametric Bayesian extension of mixtures of random effects regressions for curve clustering (Gaffney and Smyth, 2003). Like previous work, the GMT model assumes that the model order is known a priori. In Section 3.3, we present the DP-GMT model extending GMT by using a Dirichlet process prior on the mixture proportions so that the number of mixture components is adaptively determined by the data rather than being fixed explicitly.

Our main technical contribution is the inference algorithm for the GMT and DP-GMT. We develop details for the EM algorithm for the GMT model and a Variational EM algorithm for DP-GMT optimizing the maximum a posteriori (MAP) estimates for the parameters of the models. The main insights in the GMT solution are in estimating the expectation for the coupled hidden variables (the cluster identities and the task specific portion of the time series) and in solving the regularized least squares problem for a set of phase-shifted observations. In addition, for the DP-GMT, we show that the variational EM algorithm can be implemented with the same complexity as the fixed order GMT without using sampling. Thus the DP-GMT provides an efficient model selection algorithm compared to alternatives such as Bayesian Information Criterion (BIC). As a special case our algorithm yields the (Infinite) Gaussian mixture model for phase shifted time series, which may be of independent interest, and which is a generalization of the algorithms of Rebbapragada et al. (2009) and Gaffney and Smyth (2003).

Our model primarily captures regression of time series but because it is a generative model it can be used for class discovery, clustering, and classification. We demonstrate the utility of the model using several experiments with both synthetic data and real-world time series data from astrophysics. The experiments show that our model can yield superior results when compared to the single-task learning and Gaussian mixture models, especially when each individual task is sparsely and non-synchronously sampled. The DP-GMT model yields results that are competitive with model selection using BIC over the GMT model, at much reduced computational cost.

The remainder of the chapter is organized as follows. Section 3.1 defines the new generative model, Section 3.2 develops the EM algorithm for it, and the infinite mixture extension is addressed in Section 3.3. The experimental results are reported in Section 3.4. Related work is discussed in Section 3.5 and the final section concludes with a discussion and outlines ideas for future work.

3.1 Model Description

We start by formally defining the generative model, which we call *Shift-invariant Grouped mixed-effects Model* (GMT). In this model, k group effect functions are assumed to share the same GP prior characterized by $\mathcal{K}_0$. The individual effect functions are Gaussian processes with covariance function $\mathcal{K}$. The model is shown in Figure 3-2 and it is characterized by parameter set $\mathcal{M} = \{\mathcal{K}_0, \mathcal{K}, \boldsymbol{\alpha}, \{t_j\}, \sigma^2\}$ where $\boldsymbol{\alpha}$ is the vector of the mixture proportion and t_j is the phase shift for the jth time series. The generative process is as follows

1. Draw the fixed-effect functions (centers): $\bar{f}_s|\mathcal{K}_0 \sim \exp\left\{-\frac{1}{2}\|\bar{f}_s\|^2_{\mathcal{H}_0}\right\}, \quad s = 1, 2, \cdots, k$

2. For the jth time series

   - Draw $z_j|\boldsymbol{\alpha} \sim \text{Multinomial}(\boldsymbol{\alpha})$
   - Draw the random effect: $\tilde{f}^j|\mathcal{K} \sim \exp\left\{-\frac{1}{2}\|\tilde{f}^j\|^2_{\mathcal{H}}\right\}$
   - Draw $\boldsymbol{y}^j|z_j, f^j, \boldsymbol{x}^j, t_j, \sigma^2 \sim \mathcal{N}\left(f^j(\boldsymbol{x}^j), \sigma^2\mathbb{I}_j\right)$, where $f^j = \bar{f}_{z_j} * \delta_{t_j} + \tilde{f}^j$.

Additionally, denote $\mathcal{X} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^M\}$ and $\mathcal{Y} = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^M\}$, where $\boldsymbol{x}^j$ are the time points when the jth time series is sampled and $\boldsymbol{y}^j$ are the corresponding observations.

We assume that the group effect kernel $\mathcal{K}_0$ and the number of centers k are known. The assumption on $\mathcal{K}_0$ is reasonable in that, normally, we can get more information on the shape of the mean waveforms, thereby making it possible to design the kernel for $\mathcal{H}_0$. On the other hand, the individual variations are more arbi-

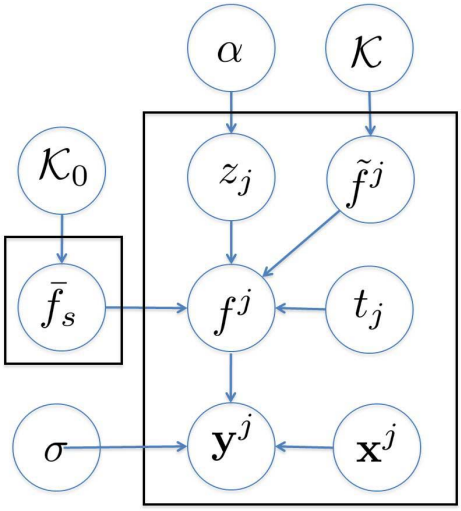

Figure 3-2: GMT: Plate graph

trary and therefore $\mathcal{K}$ is not assumed to be known. The assumption that k is known requires some form of model selection. An extension using a non-parametric Bayesian model, the *Dirichlet process* (Teh, 2010), that does not limit k is developed in Section 3.3. The group effect $\{\mathbf{f}_s\}$, individual shifts $\{t_j\}$, noise variance σ^2 and the kernel for individual variations $\mathcal{K}$ are unknown and need to be estimated. The cluster assignments $\{z_j\}$ and individual variation $\{\tilde{f}^j\}$ are treated as hidden variables. Note that one could treat $\{\mathbf{f}_s\}$ too as hidden variables, but we prefer to get a concrete estimate for these variables because of their role as the mean waveforms in our model.

The model above is a standard model for regression. We propose to use it for classification by learning a mixture model for each class and using the *Maximum A Posteriori* (MAP) probability for the class for classification. In particular, consider a training set that has L classes, where the jth instance is given by $\mathcal{Y}^j = (\boldsymbol{x}^j, \boldsymbol{y}^j, o^j) \in \mathbb{R}^{N_j} \times \mathbb{R}^{N_j} \times \{1, 2, \cdots, L\}$. Each observation $(\boldsymbol{x}^j, \boldsymbol{y}^j)$ is given a label from $\{1, 2, \cdots, L\}$. The problem is to learn the model M_ℓ for each class separately (L in total) and the classification rule for a new instance $(\boldsymbol{x}, \boldsymbol{y})$ is given by

$$o = \underset{\ell=\{1,\cdots,L\}}{\operatorname{argmax}} \Pr(\boldsymbol{y}|\boldsymbol{x}; \mathrm{M}_\ell) \Pr(\ell). \tag{3.2}$$

As we show in our experiments, the generative model can provide explanatory power for the application while giving excellent classification performance.

3.2 Parameter Estimation

Given data set $\mathcal{Y} = \{\boldsymbol{x}^j, \boldsymbol{y}^j\} = \{x_i^j, y_i^j\}, i = 1, \cdots, N_j, j = 1, \cdots, M$, the learning process aims to find the MAP estimates of the parameter set $\mathcal{M} = \{\boldsymbol{\alpha}, \{\mathbf{f}_s\}, \{t_j\}, \sigma^2, \mathcal{K}\}$

$$\mathcal{M}^* = \underset{\mathcal{M}}{\operatorname{argmax}} \left(\Pr(\mathcal{Y}|\mathcal{X}; \mathcal{M}) \times \Pr[\{\mathbf{f}_s\}; \mathcal{K}_0]\right). \tag{3.3}$$

The direct optimization of (3.3) is analytically intractable because of coupled sums that come from the mixture distribution. To solve this problem, we resort to the EM algorithm (Dempster et al., 1977).

3.2.1 EM Algorithm

The EM algorithm is an iterative method for optimizing the maximum likelihood (ML) or MAP estimates of the parameters in the context of hidden variables. Alternatively, it can be viewed as an estimation problem involving incomplete data in which each unlabeled observation in the mixture is regarded as missing its label.

Let $\mathbf{X}$ be the observed data and $\mathbf{Z}$ be the hidden variables, i.e. in the GMM case, the hidden variables indicate the class membership of the data. We are given a joint distribution of the $\Pr(\mathbf{X}, \mathbf{Z})$, governed by a set of parameters $\mathcal{O}$, our goal is to find the best parameters $\mathcal{O}^*$ that maximize $\Pr(\mathbf{X})$.

We assume that if we are given the complete data $\{\mathbf{X}, \mathbf{Z}\}$, then the ML estimate becomes significantly easier. With the hidden variable missing, the best we can do is to: 1) estimate the distribution of the hidden variables $\mathbf{Z}$ (based on the observed data $\mathbf{X}$) ; 2) find the parameters that maximize the expected complete data likelihood. This is what the EM algorithm does. More precisely, starting with an initial value of $\mathcal{O}^0$, the EM algorithm iterates between the following two steps until it converges,

- In the **E**xpectation step, one calculates the posterior distribution of the hidden variables based on the current estimate $\mathcal{O}^g$, and then calculate the following $\mathcal{Q}$ function

$$\mathcal{Q}(\mathcal{O}, \mathcal{O}_g) = \int \ln \Pr(\mathbf{X}, \mathbf{Z}|\mathcal{O}) \mathrm{d} \Pr(\mathbf{Z}|\mathbf{X}, \mathcal{O}^g).$$

- In the **M**aximization step, one finds the best parameters $\mathcal{O}^*$ that maximizes $\mathcal{Q}(\mathcal{O}, \mathcal{O}_g)$,

$$\mathcal{O}^* = \underset{\mathcal{O}}{\operatorname{argmax}}\ \mathcal{Q}(\mathcal{O}, \mathcal{O}^g),$$

and $\mathcal{O}^*$ becomes the new $\mathcal{O}^g$.

The EM algorithm is guaranteed to converge, although it may converge to a local minimum. A common approach to address this is to repeat the EM several times with different initialization and output the parameters with the maximum log likelihood.

3.2.2 Expectation step

In our case, the hidden variables are $\mathbf{z} = \{z_j\}$ (which is the same as in the standard GMM), and $\tilde{\mathbf{f}} := \{\tilde{\mathbf{f}}^j \triangleq \tilde{f}^j(\boldsymbol{x}^j)\}, j = 1, \cdots, M$. The algorithm iterates between the following expectation and maximization steps until it converges to a local maximum. In the **E**-step, we calculate

$$Q(\mathcal{M}, \mathcal{M}^g) = \mathbb{E}_{\{\mathbf{z}, \tilde{\mathbf{f}}|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g\}} \left[\log \left\{\Pr(\mathcal{Y}, \tilde{\mathbf{f}}, \mathbf{z}|\mathcal{X}; \mathcal{M}) \times \Pr[\{\mathbf{f}_s\}; \mathcal{K}_0]\right\}\right] \tag{3.4}$$

where $\mathcal{M}^g$ stands for estimated parameters from the last iteration. For our model, the difficulty comes from estimating the expectation with respect to the coupled latent variables $\{\mathbf{z}, \tilde{\mathbf{f}}\}$. In the following, we show how this can be done. First notice that,

$$\Pr(\mathbf{z}, \tilde{\mathbf{f}}|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g) = \prod_j \Pr(z_j, \tilde{\mathbf{f}}^j|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g)$$

and further that

$$\Pr(z_j, \tilde{\mathbf{f}}^j | \mathcal{X}, \mathcal{Y}; \mathcal{M}^g) = \Pr(z_j | \boldsymbol{x}^j, \boldsymbol{y}^j; \mathcal{M}^g) \times \Pr(\tilde{\mathbf{f}}^j | z_j, \boldsymbol{x}^j, \boldsymbol{y}^j; \mathcal{M}^g). \tag{3.5}$$

The first term in (3.5) can be further written as

$$\Pr(z_j | \boldsymbol{x}^j, \boldsymbol{y}^j; \mathcal{M}^g) \propto \Pr(z_j; \mathcal{M}^g) \Pr(\boldsymbol{y}^j | z_j, \boldsymbol{x}^j; \mathcal{M}^g) \tag{3.6}$$

where $\Pr(z_j; \mathcal{M}^g)$ is specified by the parameters estimated from last iteration. Since z_j is given, the second term is the marginal distribution that can be calculated using a Gaussian process regression model. In particular, denote $\mathbf{f}^j = \mathbf{f}_{z_j} * \delta_{t_j}(\boldsymbol{x}^j)$ and let $\mathbf{K}_j^g$ be the kernel matrix for the jth task using parameters from last iteration, i.e. $\mathbf{K}_j^g = (\mathcal{K}(x_i^j, x_l^j))_{il}$, the marginal distribution is

$$\boldsymbol{y}^j | z_j \sim \mathcal{N}(\mathbf{f}^j, \mathbf{K}_j^g + \sigma^2 \mathbb{I}). \tag{3.7}$$

Next consider the second term in (3.5) and recall that $\tilde{\mathbf{f}}^j = \tilde{f}^j(\boldsymbol{x}^j)$. Given z_j, there is no uncertainty about the identity of $\mathbf{f}_{z_j}$ and therefore the calculation amounts to estimating the posterior distribution under standard Gaussian process regression. In particular,

$$\begin{aligned} \boldsymbol{y}^j - \mathbf{f}^j &\sim \mathcal{N}(\tilde{f}^j(\boldsymbol{x}^j), \sigma^2 \mathbb{I}) \\ \tilde{f}^j &\sim \exp\left\{ -\frac{1}{2} \| \tilde{f}^j \|_{\mathcal{K}}^2 \right\} \end{aligned}$$

and the conditional distribution is given by

$$\tilde{\mathbf{f}}^j | z_j, \boldsymbol{x}^j, \boldsymbol{y}^j \sim \mathcal{N}(\boldsymbol{\mu}_j^g, \mathbf{C}_j^g) \tag{3.8}$$

where $\boldsymbol{\mu}_j^g$ is the posterior mean of $\tilde{\mathbf{f}}^j$

$$\boldsymbol{\mu}_j^g = \mathbf{K}_j^g (\mathbf{K}_j^g + \sigma^2 \mathbb{I})^{-1} (\boldsymbol{y}^j - \mathbf{f}^j) \tag{3.9}$$

and $\mathbf{C}_j^g$ is the posterior covariance of $\tilde{\mathbf{f}}^j$

$$\mathbf{C}_j^g = \mathbf{K}_j^g - \mathbf{K}_j^g(\mathbf{K}_j^g + \sigma^2\mathbb{I})^{-1}\mathbf{K}_j^g. \tag{3.10}$$

Since (3.6) is Multinomial and $\tilde{\mathbf{f}}^j$ is normal in (3.8), the marginal distribution of $\tilde{\mathbf{f}}^j$ is a Gaussian mixture distribution given by

$$\begin{aligned}\Pr(\tilde{\mathbf{f}}^j|\boldsymbol{x}^j, \boldsymbol{y}^j; \mathcal{M}^g) = \sum_s \Pr(z_j = s|\boldsymbol{x}^j, \boldsymbol{y}^j; \mathcal{M}^g) \\ \times \mathcal{N}\left(\mu_j^g, \mathbf{C}_j^g|z_j = s; \mathcal{M}^g\right), \quad s = 1, \cdots, k.\end{aligned}$$

To work out the concrete form of $Q(\mathcal{M}, \mathcal{M}^g)$, denote $z_{il} = 1$ if $z_i = l$ and $z_{il} = 0$ otherwise. Then the complete data likelihood can be reformulated as

$$\begin{aligned}\mathcal{L} &= \Pr(\mathcal{Y}, \mathbf{f}, \mathbf{z}; \mathcal{X}, \mathcal{M}) \\ &= \prod_{j,s}\left[\alpha_s \Pr(\boldsymbol{y}^j, \tilde{\mathbf{f}}^j|z_j = s; \mathcal{M})\right]^{z_{js}} \\ &= \prod_{j,s}\left[\alpha_s \Pr(\boldsymbol{y}^j|\tilde{\mathbf{f}}^j, z_j = s; \mathcal{M}) \Pr(\tilde{\mathbf{f}}^j; \mathcal{M})\right]^{z_{js}}\end{aligned}$$

where we have used the fact that exactly one z_{js} is 1 for each j and included the last term inside the product over s for convenience. Then (3.4) can be written as

$$Q(\mathcal{M}, \mathcal{M}^g) = -\frac{1}{2}\sum_s \|f_s\|_{\mathcal{H}_0}^2 + \mathbb{E}_{\{\mathbf{z}, \tilde{\mathbf{f}}|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g\}}\left[\log \mathcal{L}\right].$$

Denote the second term by $\widetilde{Q}$. By a version of Fubini's theorem (Stein and Shakarchi, 2005) we have

$$\begin{aligned}\widetilde{Q} &= \mathbb{E}_{\{\mathbf{z}|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g\}}\mathbb{E}_{\{\tilde{\mathbf{f}}|\mathbf{z}, \mathcal{X}, \mathcal{Y}; \mathcal{M}^g\}}\left[\log \mathcal{L}\right] \\ &= \sum_{\mathbf{z}} \Pr(\mathbf{z}|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g)\Bigg\{\sum_{j,s} z_{js} \\ &\quad \times \int d\Pr(\tilde{\mathbf{f}}^j|z_j = s) \log\left[\alpha_s \Pr(\boldsymbol{y}^j|\tilde{\mathbf{f}}^j, z_j = s; \mathcal{M}) \Pr(\tilde{\mathbf{f}}^j; \mathcal{M})\right]\Bigg\}.\end{aligned} \tag{3.11}$$

Now because the last term in (3.11) does not include any z_i, the equation can be further decomposed as

$$\begin{aligned}
\widetilde{Q} &= \sum_{j,s} \left(\sum_{\mathbf{z}} \Pr(\mathbf{z}|\mathcal{X},\mathcal{Y};\mathcal{M}^g) z_{js} \right) \\
&\qquad \times \left\{ \int d\,\Pr(\tilde{\mathbf{f}}^j|z_j=s) \log[\alpha_s \Pr(\boldsymbol{y}^j|\tilde{\mathbf{f}}^j, z_j=s;\mathcal{M}) \Pr(\tilde{\mathbf{f}}^j;\mathcal{M})] \right\} \\
&= \sum_{j,s} \gamma_{js} \int d\,\Pr(\tilde{\mathbf{f}}^j|z_j=s) \log \left[\alpha_s \Pr(\boldsymbol{y}^j|\tilde{\mathbf{f}}^j, z_j=s;\mathcal{M}) \Pr(\tilde{\mathbf{f}}^j;\mathcal{M}) \right] \\
&= \sum_{j,s} \gamma_{js} \mathbb{E}_{\{\tilde{\mathbf{f}}^j|z_j=s,\boldsymbol{x}^j,\boldsymbol{y}^j;\mathcal{M}^g\}} \left[\log \alpha_s + \log \left(\Pr(\boldsymbol{y}^j|\tilde{\mathbf{f}}^j, z_j=s;\mathcal{M}) \right) + \log \left(\Pr(\tilde{\mathbf{f}}^j;\mathcal{M}) \right) \right]
\end{aligned} \tag{3.12}$$

where

$$\gamma_{js} = \mathbb{E}[z_{js}|\boldsymbol{y}^j, \boldsymbol{x}^j; \mathcal{M}^g] = \frac{\Pr(z_j=s|\boldsymbol{x}^j, \boldsymbol{y}^j; \mathcal{M}^g)}{\sum_s \Pr(z_j=s|\boldsymbol{x}^j, \boldsymbol{y}^j; \mathcal{M}^g)} \tag{3.13}$$

can be calculated from (3.6) and (3.7) and γ_{js} can be viewed as a fractional label indicating how likely the jth task is to belong to the sth group. Recall that $\Pr(\boldsymbol{y}^j|\tilde{\mathbf{f}}^j, z_j=s)$ is a normal distribution given by

$$\mathcal{N}([\mathbf{f}_{z_j} * \delta_{t_j}](\boldsymbol{x}^j) + \tilde{\mathbf{f}}^j, \sigma^2 \mathbb{I}),$$

and $\Pr(\tilde{\mathbf{f}}^j;\mathcal{M})$ is a standard multivariate Gaussian distribution determined by its prior $\mathcal{N}(\mathbf{0}, \mathbf{K}_j)$. Using these facts and (3.12), $Q(\mathcal{M}, \mathcal{M}^g)$ can be re-formulated as

$$\begin{aligned}
Q(\mathcal{M}, \mathcal{M}^g) = &-\frac{1}{2} \sum_s \|\mathbf{f}_s\|^2_{\mathcal{H}_0} - \sum_j n_j \log \sigma + \sum_{j,s} \gamma_{js} \log \alpha_s \\
&- \frac{1}{2\sigma^2} \sum_{j,s} \gamma_{js} \mathbb{E}_{\{\tilde{\mathbf{f}}^j|z_j=s,\boldsymbol{x}^j,\boldsymbol{y}^j;\mathcal{M}^g\}} \left[\|\boldsymbol{y}^j - [\mathbf{f}_s * \delta_{t_j}](\boldsymbol{x}^j) - \tilde{\mathbf{f}}^j\|^2 \right] \\
&- \frac{1}{2} \sum_j \log |\mathbf{K}_j| - \frac{1}{2} \sum_{j,s} \gamma_{js} \mathbb{E}_{\{\tilde{\mathbf{f}}^j|z_j=s,\boldsymbol{x}^j,\boldsymbol{y}^j;\mathcal{M}^g\}} \left((\tilde{\mathbf{f}}^j)^{\mathrm{T}} \mathbf{K}_j^{-1} \tilde{\mathbf{f}}^j \right) + \mathrm{CONST}
\end{aligned} \tag{3.14}$$

We next develop explicit closed forms for the remaining expectations. For the first, note that for $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and a constant vector $\boldsymbol{a}$,

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{a}-\boldsymbol{x}\|^2] &= \mathbb{E}[\|\boldsymbol{a}\|^2 - 2\langle \boldsymbol{a}, \boldsymbol{x}\rangle + \|\boldsymbol{x}\|^2] \\ &= \|\boldsymbol{a}\|^2 - 2\langle \boldsymbol{a}, \mathbb{E}[\boldsymbol{x}]\rangle + \mathbb{E}[\boldsymbol{x}]^2 + \mathrm{Tr}(\boldsymbol{\Sigma}) = \|\boldsymbol{a}-\boldsymbol{\mu}\|^2 + \mathrm{Tr}(\boldsymbol{\Sigma}).\end{aligned}$$

Therefore the expectation is

$$\begin{aligned}\frac{1}{2\sigma^2}\sum_{j,s}\gamma_{js}\cdot \mathbb{E}_{\{\tilde{\mathbf{f}}^j|z_j=s,\boldsymbol{x}^j,\boldsymbol{y}^j;\mathcal{M}^g\}}\left[\|\boldsymbol{y}^j - [\mathbf{f}_s * \delta_{t_j}](\boldsymbol{x}^j) - \tilde{\mathbf{f}}^j\|^2\right] &= \frac{1}{2\sigma^2}\sum_j \mathrm{Tr}(\mathbf{C}_j^g) \\ &+ \frac{1}{2\sigma^2}\sum_{j,s}\gamma_{js}\left(\|\boldsymbol{y}^j - [f_s * \delta_{t_j}](\boldsymbol{x}^j) - \mu_{js}^g\|^2\right)\end{aligned} \tag{3.15}$$

where $\mu_{js}^g = \mathbb{E}_{\{\tilde{\mathbf{f}}^j|z_j=s,\boldsymbol{x}^j,\boldsymbol{y}^j;\mathcal{M}^g\}}[\tilde{\mathbf{f}}^j]$ is as in (3.9) where we set $z_j = s$ explicitly. For the second expectation we have

$$\begin{aligned}\mathbb{E}_{\{\tilde{\mathbf{f}}^j|z_j=s,\boldsymbol{x}^j,\boldsymbol{y}^j;\mathcal{M}^g\}}\left(\mathbf{f}_j^{\mathrm{T}}\mathbf{K}_j^{-1}\tilde{\mathbf{f}}^j\right) &= \mathbb{E}_{\{\tilde{\mathbf{f}}^j|z_j=s,\boldsymbol{x}^j,\boldsymbol{y}^j;\mathcal{M}^g\}}\left[\mathrm{Tr}\left((\tilde{\mathbf{f}}^j)^{\mathrm{T}}\mathbf{K}_j^{-1}\tilde{\mathbf{f}}^j\right)\right] \\ &= \mathbb{E}_{\{\tilde{\mathbf{f}}^j|z_j=s,\boldsymbol{x}^j,\boldsymbol{y}^j;\mathcal{M}^g\}}\left[\mathrm{Tr}\left(\mathbf{K}_j^{-1}\tilde{\mathbf{f}}^j(\tilde{\mathbf{f}}^j)^{\mathrm{T}}\right)\right] \\ &= \mathrm{Tr}\left(\mathbb{E}_{\{\tilde{\mathbf{f}}^j|z_j=s,\boldsymbol{x}^j,\boldsymbol{y}^j;\mathcal{M}^g\}}[\mathbf{K}_j^{-1}\tilde{\mathbf{f}}^j(\tilde{\mathbf{f}}^j)^{\mathrm{T}}]\right) \\ &= \mathrm{Tr}\left(\mathbf{K}_j^{-1}(\mathbf{C}_j^g + \mu_{js}^g(\mu_{js}^g)^{\mathrm{T}})\right).\end{aligned}$$

3.2.3 Maximization step

In this step, we aim to find

$$\mathcal{M}^* = \underset{\mathcal{M}}{\mathrm{argmax}}\; Q(\mathcal{M}, \mathcal{M}^g)$$

and use $\mathcal{M}^*$ to update the model parameters. Using the results above this can be decomposed into three separate optimization problems as follows:

$$\mathcal{M}^* = \underset{\mathcal{M}}{\mathrm{argmax}}\left\{Q_1((\{\mathbf{f}_s\},\{\delta_{t_j}\},\sigma)) + Q_2(\mathcal{K}) + \sum_{j,s}\gamma_{js}\log\alpha_s\right\}.$$

That is, $\boldsymbol{\alpha}$ can be estimated easily using its separate term, Q_1 is only a function of $(\{\mathbf{f}_s\}, \{t_j\}, \sigma)$ and Q_2 depends only on $\mathcal{K}$, and we have

$$\begin{aligned} Q_1(\{\mathbf{f}_s\}, \{t_j\}, \sigma^2) = \frac{1}{2}\sum_s \|\mathbf{f}_s\|^2_{\mathcal{K}_0} + \sum_j n_j \log\sigma + \frac{1}{2\sigma^2}\sum_j \mathrm{Tr}(\mathbf{C}_j^g) \\ + \frac{1}{2\sigma^2}\sum_{j,s}\gamma_{js}\left(\|\boldsymbol{y}^j - [f_s * \delta_{t_j}](\boldsymbol{x}^j) - \mu_{js}^g\|^2\right) \end{aligned} \tag{3.16}$$

and

$$Q_2(\mathcal{K}) = -\frac{1}{2}\sum_j \log|\mathbf{K}_j| - \frac{1}{2}\sum_{j,s}\gamma_{js}\mathrm{Tr}\left(\mathbf{K}_j^{-1}(\mathbf{C}_j^g + \mu_{js}^g(\mu_{js}^g)^{\mathrm{T}})\right). \tag{3.17}$$

The optimizations for Q_1 and Q_2 are described separately in the following two subsections.

Optimize Q_1: Learning $\{\mathbf{f}_s\}, \{t_j\}, \sigma^2$

To optimize (3.16), we assume first that σ is given. To simplify the optimization, we introduce the following heuristic: estimate separate time shifts t_j for each center f_s. Intuitively this can speed up convergence in early iterations. Considering a time series whose membership has not yet been determined, we can expect the estimate of its phase shift to be affected by all centers leading to an inaccurate estimation. In turn, this will slow down the identification of its cluster membership. With the heuristic, the phase shift of a star in its own true class will be better estimated leading to better identification of its cluster membership. Considering the limiting case, when membership has converged to be close to zero or one for each cluster, the irrelevant t_j values (for cluster with membership close to 0) do not affect the result, and we get individual t_j estimates from their cluster. In practice, we get improved convergence with this slightly modified model.

Using this heuristic, estimating $\{\mathbf{f}_s\}, \{t_j\}$ decouples into k sub-problems, finding sth group effect $\mathbf{f}_s$ and the corresponding time shift $\{t_j\}$. Denoting the residual

$\tilde{\mathbf{y}}^j = \boldsymbol{y}^j - \mu_{js}$, where $\mu_{js} = \mathbb{E}[\tilde{\mathbf{f}}^j | \boldsymbol{y}^j, z_j = s]$, the problem becomes

$$\underset{f \in \mathcal{H}_0, t_1, \cdots, t_M \in [0,T)}{\operatorname{argmin}} \left\{ \frac{1}{2\sigma^2} \sum_j \gamma_{js} \sum_{i=1}^{n_j} (\tilde{\mathbf{y}}_i^j - [f * \delta_{t_j}](\boldsymbol{x}_i^j))^2 + \frac{1}{2} \|f\|_{\mathcal{H}_0}^2 \right\}. \tag{3.18}$$

Note that different $\boldsymbol{x}^j, \boldsymbol{y}^j$ have different dimensions n_j and they are not assumed to be sampled at regular intervals. For further development, following Pillonetto et al. (2010), it is useful to introduce the distinct vector $\breve{\boldsymbol{x}} \in \mathbb{R}^{\mathbb{N}}$ whose component are the distinct elements of $\mathcal{X}$. For example if $\boldsymbol{x}^1 = [1,2,3]^T, \boldsymbol{x}^2 = [2,3,4,5]^T$, then $\breve{\boldsymbol{x}} = [1,2,3,4,5]^T$. For the jth task, let the binary matrix C^k be such that

$$\boldsymbol{x}^j = C^j \cdot \breve{\boldsymbol{x}}, \quad f(\boldsymbol{x}^j) = C^j \cdot f(\breve{\boldsymbol{x}}).$$

That is, C^j extracts the values corresponding to the jth task from the full vector. If $\{t_j\}$ are fixed, then the optimization in (3.18) is standard and the representer theorem (Scholkopf and Smola, 2002) gives the form of the solution as

$$f(\cdot) = \sum_{i=1}^{\mathbb{N}} c_i \mathcal{K}_0(\breve{x}_i, \cdot). \tag{3.19}$$

Denoting the kernel matrix as $\mathfrak{K} = \mathcal{K}_0(\breve{x}_i, \breve{x}_j), i, j = 1, \cdots, \mathbb{N}$, and $\mathbf{c} = [c_1, \cdots, c_{\mathbb{N}}]^T$, we get $f(\breve{\boldsymbol{x}}) = \mathfrak{K}\mathbf{c}$. To simplify the optimization we assume that $\{t_j\}$ can only take values in the discrete space $\{\tilde{t}_1, \cdots, \tilde{t}_L\}$, that is, $t_j = \tilde{t}_i$, for some $i \in 1, 2, \cdots, L$ (e.g., a fixed finite fine grid), where we always choose $\tilde{t}_1 = 0$. Therefore, we can write $\left[f * \delta_{t_j} \right] (\breve{\boldsymbol{x}}) = \widetilde{\mathcal{K}}_{t_j}^{\mathrm{T}} \mathbf{c}$, where $\widetilde{\mathcal{K}}_{t_j}$ is $\mathcal{K}_0(\breve{\boldsymbol{x}}, [(\breve{\boldsymbol{x}} - \tilde{t}_j) \mod T])$. Accordingly, (3.18) can be reduced to

$$\underset{\mathbf{c} \in \mathbb{R}^{\mathbb{N}}, t_1, \cdots, t_j \in \{\tilde{t}_i\}}{\operatorname{argmin}} \left\{ \sum_j \gamma_{js} \|\tilde{\mathbf{y}}^j - C^j \cdot \widetilde{\mathcal{K}}_{t_j}^{\mathrm{T}} \mathbf{c}\|^2 + \frac{1}{2} \mathbf{c}^{\mathrm{T}} \mathfrak{K} \mathbf{c} \right\}. \tag{3.20}$$

To solve this optimization, we follow a cyclic optimization approach where we alternate between steps of optimizing f and $\{t_j\}$ respectively,

- At step ℓ, optimize equation (3.20) with respect to $\{t_j\}$ given $\mathbf{c}^{(\ell)}$. Since $\mathbf{c}^{(\ell)}$ is

known, it follows immediately that (3.20) decomposes into M independent tasks, where for the jth task we need to find $t_j^{(\ell)}$ such that $C^j \widetilde{\mathcal{K}}^{\mathrm{T}}_{t_j^{(\ell)}} \mathbf{c}$ is closest to $\tilde{\mathbf{y}}^j$ under the Euclidean distance. A brute force search with time complexity $\mathcal{O}(\mathbb{N}L)$ yields the optimal solution. If the time series are synchronously sampled (i.e. $C^j = \mathbb{I}, j = 1, \cdots, M$), this is equivalent to finding the shift τ corresponding the *cross-correlation*, defined as

$$\mathcal{C}(\mathbf{u}, \mathbf{v}) = \max_{\tau} \langle \mathbf{u}, \mathbf{v}_{+\tau} \rangle \tag{3.21}$$

where $\mathbf{u} = \mathfrak{K}\mathbf{c}$ and $\mathbf{v} = \tilde{\mathbf{y}}^j$ and $\mathbf{v}_{+\tau}$ refers to the vector $\mathbf{v}$ right shifted by τ positions, and where positions are shifted modulo $\mathbb{N}$. Furthermore, as shown by Protopapas et al. (2006), if every $\boldsymbol{x}^j$ has regular time intervals, we can use the convolution theorem to find the same value in $\mathcal{O}(\mathbb{N} \log \mathbb{N})$ time, that is

$$t_j^{(\ell)} = \underset{\tau}{\operatorname{argmax}} \left(\mathscr{F}^{-1} \left[\mathscr{U} \cdot \widehat{\mathscr{V}} \right] (\tau) \right) \tag{3.22}$$

where $\mathscr{F}^{-1}[\cdot]$ denotes inverse Fourier transform, $\cdot$ indicates point-wise multiplication; $\mathscr{U}$ is the Fourier transform of $\mathbf{u}$ and $\widehat{\mathscr{V}}$ is the complex conjugate of the Fourier transform of $\mathbf{v}$.

- At step $\ell + 1$, optimize equation (3.20) with respect to $\mathbf{c}^{(\ell+1)}$ given $t_1^{(\ell)}, \cdots, t_M^{(\ell)}$. For the jth task, since $t_j^{(\ell)}$ is known, denote $C^j \widetilde{\mathcal{K}}^{\mathrm{T}}_{t_j^{(\ell)}}$ as $\mathfrak{M}_j^{(\ell)}$. The regularized least square problem can be reformulated as

$$\underset{\mathbf{c} \in \mathbb{R}^{\mathbb{N}}}{\operatorname{argmin}} \left\{ \sum_j \gamma_{js} \| \tilde{\mathbf{y}}^j - \mathfrak{M}_j^{(\ell)} \mathbf{c} \|^2 + \frac{1}{2} \mathbf{c}^T \mathfrak{K} \mathbf{c} \right\}. \tag{3.23}$$

Taking derivatives of (3.23), we see that the new $\mathbf{c}^{(\ell+1)}$ value is obtained by solving the following linear system

$$-2 \sum_j \gamma_{js} \cdot (\mathfrak{M}_j^{(\ell)})^{\mathrm{T}} \left(\tilde{\mathbf{y}}^j - \mathfrak{M}_j^{(\ell)} \cdot \mathbf{c} \right) + \mathfrak{K}\mathbf{c} = 0. \tag{3.24}$$

Obviously, each step decreases the value of the objective function and therefore the algorithm will converge.

Given the estimates of $\{\mathbf{f}_s\}, \{t_j\}$, the optimization for σ^2 is given by

$$\sigma^* = \underset{\sigma \in \mathbb{R}}{\operatorname{argmin}} \left\{ \sum_j n_j \log \sigma + \frac{1}{2\sigma^2} \sum_j \operatorname{Tr}(\mathbf{C}_j^g) \right. \\ \left. \frac{1}{2\sigma^2} \sum_{j,s} \gamma_{js} \left(\|\boldsymbol{y}^j - [\mathbf{f}_s^* * \delta_{t_j^*}](\boldsymbol{x}^j) - \boldsymbol{\mu}_{js}\|^2 \right) \right\} \tag{3.25}$$

where $\{\mathbf{f}^*\}$ and $\{t_j^*\}$ are obtained from the previous optimization steps. Let $R = \sum_j \operatorname{Tr}(\mathbf{C}_j^g) + \sum_{j,s} \gamma_{js} \left(\|\boldsymbol{y}^j - [\mathbf{f}_s^* * \delta_{t_j}](\boldsymbol{x}^j) - \boldsymbol{\mu}_{js}\|^2 \right)$. Then it is easy to see that $(\sigma^*)^2 = R / \sum_j n_j$.

Optimize Q_2: Learning the Kernel for the Random Effect

Lu et al. (2008) have already shown how to optimize the kernel function in a similar context. Here we provide some of the details for completeness. If the kernel function $\mathcal{K}$ admits a parametric form with parameter $\mathcal{O}$, for example the RBF kernel as in (2.10) where $\mathcal{O} = \{a, \sigma^2\}$, then the optimization of the kernel $\mathcal{K}$ amounts to finding $\mathcal{O}^*$ such that

$$\mathcal{O}^* = \underset{\mathcal{O}}{\operatorname{argmax}} \left\{ -\frac{1}{2} \sum_j \log |(\mathbf{K}_j; \mathcal{O})| - \frac{1}{2} \sum_{j,s} \gamma_{js} \operatorname{Tr} \left((\mathbf{K}_j; \mathcal{O})^{-1} (\mathbf{C}_j^g + \boldsymbol{\mu}_{js}^g (\boldsymbol{\mu}_{js}^g)^{\mathrm{T}}) \right) \right\}. \tag{3.26}$$

It is easy to see that the gradient of the right hand side of (3.26) is

$$-\frac{1}{2} \sum_j \operatorname{Tr} \left(\mathbf{K}_j \frac{\partial \mathbf{K}_j}{\partial \mathcal{O}} \right) - \frac{1}{2} \sum_{j,s} \gamma_{js} \operatorname{Tr} \left(\mathbf{K}_j^{-1} \frac{\partial \mathbf{K}_j}{\partial \mathcal{O}} \mathbf{K}_j^{-1} (\mathbf{C}_j^g + \boldsymbol{\mu}_{js}^g (\boldsymbol{\mu}_{js}^g)^{\mathrm{T}}) \right). \tag{3.27}$$

Therefore, any optimization method, e.g. conjugate gradients can be utilized to find the optimal parameters. Notice that given the inverse of kernel matrix $\{\mathbf{K}_j\}$, the computation of the derivative requires $|\mathcal{O}| \cdot \sum N_j^2$ steps. The parametric form of the kernel is a prerequisite to perform the regression task when examples are

not sampled synchronously as in our development above.

If the data is synchronously sampled, for classification tasks we only need to find the kernel matrix $\mathbf{K}$ for the given sample points and the optimization problem can be rewritten as

$$\mathbf{K}^* = \underset{\mathbf{K}}{\operatorname{argmax}} \left\{ -\frac{1}{2}\sum_j \log|\mathbf{K}| - \frac{1}{2}\sum_{j,s} \gamma_{js} \operatorname{Tr}\left(\mathbf{K}^{-1}(\mathbf{C}_j^g + \boldsymbol{\mu}_{js}^g(\boldsymbol{\mu}_{js}^g)^{\mathrm{T}})\right) \right\}. \tag{3.28}$$

Similar to maximum likelihood estimation for multivariate Gaussian distribution, the solution is

$$\mathbf{K}^* = \frac{1}{M}\sum_{j,s} \gamma_{js}(\mathbf{C}_j^g + \boldsymbol{\mu}_{js}^g(\boldsymbol{\mu}_{js}^g)^{\mathrm{T}}). \tag{3.29}$$

In our experiments, we use both approaches where for the parametric form we use the RBF kernel as outlined above.

3.2.4 Algorithm Summary

The various steps in our algorithm and their time complexity are summarized in Algorithm 1.

Once the model parameters $\mathcal{M}$ are learned (or if they are given in advance), we can use the model to perform regression or classification tasks. The following summarizes the procedures used in our experiments.

- **Regression:** To predict a new sample point for an existing task (task j) we calculate its most likely cluster assignment z_j and then predict the y value based on this cluster. Concretely, z_j is determined by

$$z_j = \underset{s=\{1,\cdots,k\}}{\operatorname{argmax}} \left[\Pr(z_j = s|\boldsymbol{x}^j, \boldsymbol{y}^j; \mathcal{M})\right] \tag{3.30}$$

  and given a new data point x, the prediction y is given by

$$y = \left[\mathbf{f}_{z_j} * \delta_{t_j}\right](x) + \tilde{f}^j(x).$$

Algorithm 1 EM ALGORITHM FOR SHIFT-INVARIANT GMT

Initialize $\{f_s^{(0)}\}, \{t_j^{(0)}\}, \alpha^{(0)}$ and $\mathcal{K}^{(0)}$.
repeat
    Calculate $\mathbf{K}_j^{(t)}$ according to $\boldsymbol{x}^j, \mathcal{K}^{(t-1)}$. The time complexity for constructing kernel are $\mathcal{O}(\sum N_j^2)$ and $\mathcal{O}(1)$ in parametric and nonparametric case respectively.
    Calculate γ_{js} according to (3.13). For each task, we need to invert the covariance matrix in the marginal distribution and then calculate the likelihood, thus the time complexity is $\mathcal{O}(\sum N_j^3)$.
    for all s such that $0 \leq s \leq k$ **do**
        Update $\alpha^{(t)}$ such that $\alpha_s^{(t)} = \sum_j \gamma_{js}/M$.
        repeat
            Update $\{t_j\}$ w.r.t. cluster s such that $t_j \in \{\tilde{t}_1, \cdots, \tilde{t}_L\}$ and minimize $\|\tilde{\mathbf{y}}^j - C^j \cdot \widetilde{\mathcal{K}}_{t_j}^{\mathrm{T}} \mathbf{c}_s^{(0)}\|^2$. The time complexity is $\mathcal{O}(L\mathbb{IN})$ as discussed above.
            Update $\mathbf{c}_s^{(t+1)}$ by solving linear system (3.24), which requires $\mathcal{O}(\mathbb{IN}^3)$.
        until converges **or** reach the iteration limit
    end for
    Update $\sigma^{(t+1)}$ according to (3.25).
    Update the parameters of the kernel or the kernel matrix directly via optimizing (3.26) or using the closed-form solution (3.29) for $\mathbf{K}$. In the former case, a gradient based optimizer can be used with time complexity $\mathcal{O}(\sum N_j^2)$ for each iteration; while in the later case, the estimation only requires $\mathcal{O}(kM\mathbb{IN})$.
until converges or reach the iteration limit

- **Classification:** For classification, we get a new time series and want to predict its label. Recall from Section 3.1 that we learn a separate model for each class and predict using

$$o = \underset{\ell=\{1,\cdots,L\}}{\operatorname{argmax}} \Pr(\boldsymbol{y}|\boldsymbol{x}; \mathrm{M}_\ell) \Pr(\ell).$$

  In this context, $\Pr(\ell)$ is estimated by the frequencies of each class and the likelihood portion is given by first finding the best time shift t for the new

time series and then calculating the likelihood according to

$$\Pr(\boldsymbol{y}|\boldsymbol{x}; M_\ell) = \sum_z \Pr(z|\mathcal{M}_\ell) \Pr(\boldsymbol{y}|z, \boldsymbol{x}; \mathcal{M}_\ell) \tag{3.31}$$

where $\mathcal{M}_\ell$ is the learned parameter set and the second term is calculated via (3.7).

3.3 Infinite GMT

In this section we develop an extension of the model removing the assumption that the number of centers k is known in advance.

3.3.1 Dirichlet Processes Basics

We start by reviewing basic concepts for Dirichlet processes (Blei and Jordan, 2006). Suppose we have the following independent identically distributed (i.i.d.) data,

$$x_1, x_2, \cdots, x_n \sim \mathcal{F}$$

where $\mathcal{F}$ is an unknown distribution that needs to be inferred from $\{x_i\}$. A Parametric Bayesian approach assumes that $\mathcal{F}$ is given by a parametric family $\mathcal{F}_\mathcal{O}$ and the parameters $\mathcal{O}$ follow a certain distribution that comes from our prior belief. However, this assumption has limitations both in the scope and the type of inferences that can be performed. Instead, the nonparametric Bayesian approach places a prior distribution on the distribution $\mathcal{F}$ directly. The Dirichlet process (DP) is used for such a purpose. The DP is parameterized by a base distribution G_0 and a positive scaling parameter α (or concentration parameter). A random measure G is distributed according to a DP with base measure G_0 and scaling parameter α if for all finite measurable partitions $\{B_i\}, i = 1, \ldots, k$,

$$(G(B_1), G(B_2), \cdots, G(B_k)) \sim \mathrm{Dir}(\alpha G_0(B_1), \alpha G_0(B_2), \cdots, \alpha G_0(B_k)),$$

where $\text{Dir}(\cdot)$ is the Dirichlet distribution. An important property of DP is the fact that a sample G from a DP is almost surely a discrete measure (Blei and Jordan, 2006). This is illustrated in the concrete construction below.

The Dirichlet process mixture model extends this setting, where the DP is used as a nonparametric prior in a hierarchical Bayesian specification. More precisely,

$$\begin{aligned} G|\{\alpha, G_0\} &\sim \mathcal{DP}(\alpha, G_0) \\ \eta_n|G &\sim G \qquad n = 1, 2, \ldots \\ x_n|\eta_n &\sim f(x_n|\eta_n) \end{aligned}$$

where f is some probability density function that is parameterized by η. Data generated from this model can be naturally partitioned according to the distinct values of the parameter η_n. Hence, the DP mixture can be interpreted as a mixture model where the number of mixtures is flexible and grows as the new data is observed. Alternatively, we can view the infinite mixture model as the limit of the finite mixture model. Consider the Bayesian finite mixture model with a symmetric Dirichlet distribution as the prior of the mixture proportions. When the number of mixtures k goes to infinity, the Dirichlet distribution becomes a Dirichlet process (see Neal, 2000).

Sethuraman (1994) provides a more explicit construction of the DP which is called the *stick-breaking construction* (SBC). Given $\{\alpha, G_0\}$, we have two collections of random variables $\mathbf{v}_i \sim \text{Beta}(1, \alpha)$ and $\eta_i^* \sim G_0, \quad i = \{1, 2, \cdots, \}$. The SBC of G is

$$\begin{aligned} \pi_i(\mathbf{v}) &= v_i \prod_{j=1}^{i-1} (1 - v_j) \\ G &= \sum_{i=1}^{\infty} \pi_i(\mathbf{v}) \delta_{\eta_i^*}. \end{aligned}$$

Here we see explicitly that G is a discrete measure. If we set $v_K = 1$ for some K,

then we get a truncated approximation to the DP

$$G = \sum_{i=1}^{K} \pi_i(\mathbf{v})\delta_{\eta_i^*}.$$

Ishwaran and James (2001) show that when selecting the truncation level K appropriately, the truncated DP behaves very similarly to the original DP.

3.3.2 The DP-GMT Model

In this section, we extend our model by modeling the mixture proportions using a DP prior. The plate graph is shown in Figure 3-3. Under the SBC, the generative process is as follows

1. Draw $\mathbf{v}_s|\boldsymbol{\alpha} \sim \text{Beta}(1, \boldsymbol{\alpha}), s = \{1, 2, \ldots\}$
2. Draw $\mathbf{f}_s|\mathcal{K}_0 \sim \exp\left\{-\frac{1}{2}\|\mathbf{f}_s\|^2_{\mathcal{H}_0}\right\}, s = \{1, 2, \ldots\}$
3. For the jth time series
   (a) Draw $z_j|\{v_1, v_2, \ldots\} \sim \text{Discrete}(\pi(\mathbf{v}))$, where $\pi_s(\mathbf{v}) = v_s \prod_{i=1}^{s-1}(1 - v_i)$;
   (b) Draw $\tilde{f}^j|\mathcal{K} \sim \exp\left\{-\frac{1}{2}\|\tilde{f}^j\|^2_{\mathcal{H}}\right\}$;
   (c) Draw $\boldsymbol{y}^j|z_j, f^j, \boldsymbol{x}^j, t_j, \sigma^2 \sim \mathcal{N}\left(f^j(\boldsymbol{x}^j), \sigma^2\mathbb{I}\right)$, where $f^j = \mathbf{f}_{z_j} * \delta_{t_j} + \tilde{f}^j$.

In this model, the concentration parameter $\boldsymbol{\alpha}$ is assumed to be known. As in Section 3.2, the inference task is to find the MAP estimates of the parameter set $\mathcal{M} = \{\{\mathbf{f}_s\}, \{t_j\}, \sigma^2, \mathcal{K}\}$. Notice that in contrast with the previous model, the mixture proportion are not estimated here. To perform the inference, we must consider another set of hidden variables $\mathbf{v} = \{v_i\}$ in addition to $\tilde{\mathbf{f}}$ and $\mathbf{z}$. However, calculating the posterior of the hidden variables is intractable, thus the variational EM algorithm (e.g., Bishop, 2006) is used to perform the approximate inference. The next section briefly reviews the variational EM algorithms and the following section provides the concrete algorithm for DP-GMT inference.

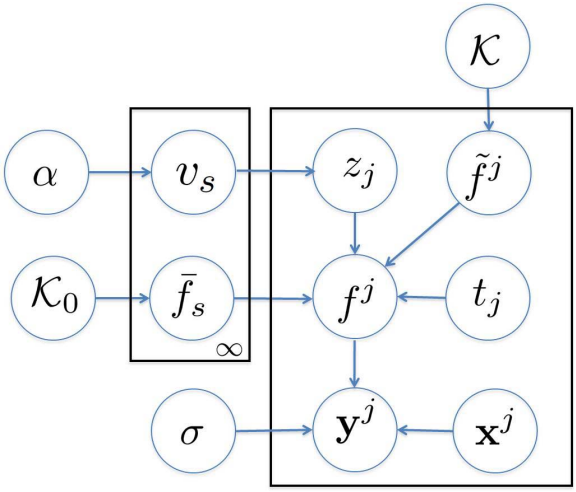

Figure 3-3: DP-GMT: The Plate graph of an infinite mixture of shift-invariant mixed-effects GP model.

3.3.3 Variational EM

As in Section 3.2.1, we denote the $\mathbf{X}$ as the observed data and $\mathbf{Z}$ as the hidden variables. We are given a joint distribution $\Pr(\mathbf{X}, \mathbf{Z})$, governed by a set of parameters $\boldsymbol{\mathcal{O}}$. Again, our goal is to maximize the likelihood function $\Pr(\mathbf{X}|\boldsymbol{\mathcal{O}})$ and we assume that optimizing $\Pr(\mathbf{X}, \mathbf{Z})$ is much easier than optimizing $\Pr(\mathbf{X})$.

Recall that in the EM algorithm (see Section 3.2.1), we need to calculate the posterior distribution over the hidden variables $\Pr(\mathbf{Z}|\mathbf{X}, \boldsymbol{\mathcal{O}})$ and evaluate expectations with respect to this distribution. However, in practice, for many models of interest, it is infeasible to calculate the exact posterior distribution. Thus, we need some kind of approximate inference, i.e., approximate the true posterior $\Pr(\mathbf{Z}|\mathbf{X}, \boldsymbol{\mathcal{O}})$. To this end, we introduce a so-called variational distribution $q(\mathbf{Z})$ that is tractable and our goal is to approximate the true posterior distribution. The KL divergence is widely used to measure the distance between two distributions. As it turns out, minimizing the KL divergence between $q(\mathbf{Z})$ and $\Pr(\mathbf{Z}|\mathbf{X}, \boldsymbol{\mathcal{O}})$ amounts to maximizing a lower bound on the true likelihood $\Pr(\mathbf{X}|\boldsymbol{\mathcal{O}})$. To see this, notice that the

following decomposition holds

$$\begin{aligned}\ln \Pr(\mathbf{X}|\boldsymbol{\mathcal{O}}) &= \int q(\mathbf{Z}) \ln \left[\frac{\Pr(\mathbf{X},\mathbf{Z}|\boldsymbol{\mathcal{O}})}{q(\mathbf{Z})}\right] \mathrm{d}\mathbf{Z} - \int q(\mathbf{Z}) \ln \left[\frac{\Pr(\mathbf{Z}|\mathbf{X},\boldsymbol{\mathcal{O}})}{q(\mathbf{Z})}\right] \mathrm{d}\mathbf{Z} \\ &= \mathcal{L}(q,\boldsymbol{\mathcal{O}}) + \mathbf{KL}(q(\mathbf{Z})||\Pr(\mathbf{Z}|\mathbf{X},\boldsymbol{\mathcal{O}})).\end{aligned} \tag{3.32}$$

As the KL divergence is greater or equal to 0, we have the $\mathcal{L}(q,\boldsymbol{\mathcal{O}})$ is a lower bound on the log likelihood. This is normally called the variational lower bound. Variational EM optimizes $\mathcal{L}(q,\boldsymbol{\mathcal{O}})$ iteratively via the following two steps,

- In the Variational **E**xpectation step, one finds the best variational distribution (in a predefined family of distributions) of the hidden variables based on the current estimate $\boldsymbol{\mathcal{O}}^g$, i.e.,

$$\begin{aligned}q^*(\mathbf{Z}) &= \underset{q}{\operatorname{argmax}}\ \mathcal{L}(q,\boldsymbol{\mathcal{O}}^g) \\ &= \underset{q}{\operatorname{argmin}}\ \mathbf{KL}(q(\mathbf{Z})||\Pr(\mathbf{Z}|\mathbf{X},\boldsymbol{\mathcal{O}}^g))\end{aligned}$$

  The second equation holds because $\ln \Pr(\mathbf{X}|\boldsymbol{\mathcal{O}})$ is independent of q, thus the largest value of $\mathcal{L}$ occurs when the **KL** divergence is at its minimum (0 when the variational distribution equals the true posterior).

- In the Variational **M**aximization step, one finds the best parameters $\boldsymbol{\mathcal{O}}^*$ such that

$$\boldsymbol{\mathcal{O}}^* = \underset{\boldsymbol{\mathcal{O}}}{\operatorname{argmax}}\ \mathcal{L}(q,\boldsymbol{\mathcal{O}}),$$

  and $\boldsymbol{\mathcal{O}}^*$ becomes the new $\boldsymbol{\mathcal{O}}^g$.

The MAP extension is straightforward. When the variational distribution is chosen to be the true posterior distribution, the KL divergence term vanishes and the variational EM reduces to the EM algorithm (see Section 3.2.1).

3.3.4 Inference of DP-GMT

In our case, the parameters are $\mathcal{M} = \{\{\mathbf{f}_s\}, \{t_j\}, \sigma^2, \mathcal{K}\}$ and the hidden variables are $\{\tilde{\mathbf{f}}, \mathbf{v}, \mathbf{z}\}$. In a high level, the algorithm can be summarized as follows:

- **Variational E-Step** Choose a family $\mathcal{G}$ of variational distributions $q(\tilde{\mathbf{f}}, \mathbf{v}, \mathbf{z})$ and find the distribution q^* that minimizes the Kullback-Leibler (**KL**) divergence between the posterior distribution and the proposed distribution given the current estimate of parameters, i.e.,

$$q^*(\tilde{\mathbf{f}}, \mathbf{v}, \mathbf{z}; \mathcal{M}^g) = \underset{q \in \mathcal{G}}{\operatorname{argmin}} \, \mathbf{KL}\left(q(\tilde{\mathbf{f}}, \mathbf{v}, \mathbf{z}))||\Pr(\tilde{\mathbf{f}}, \mathbf{v}, \mathbf{z}|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g)\right) \tag{3.33}$$

  where

$$\mathbf{KL}\left(q(\tilde{\mathbf{f}}, \mathbf{v}, \mathbf{z})||\Pr(\tilde{\mathbf{f}}, \mathbf{v}, \mathbf{z}|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g)\right) = \int \log\left[\frac{\Pr(\tilde{\mathbf{f}}, \mathbf{v}, \mathbf{z}|\mathcal{X}, \mathcal{Y}; \mathcal{M}^g)}{q(\tilde{\mathbf{f}}, \mathbf{v}, \mathbf{z})}\right] \mathrm{d}q(\tilde{\mathbf{f}}, \mathbf{v}, \mathbf{z}).$$

- **Variational M-Step** Optimize the parameter set $\mathcal{M}$ such that

$$\mathcal{M}^* = \underset{\mathcal{M}}{\operatorname{argmax}} \, Q(\mathcal{M}, \mathcal{M}^g)$$

  where

$$Q(\mathcal{M}, \mathcal{M}^g) = \mathbb{E}_{q^*(\mathbf{z}, \mathbf{f}, \mathbf{v}; \mathcal{M}^g)}\left[\log\left\{\Pr(\mathcal{Y}, \tilde{\mathbf{f}}, \mathbf{z}, \mathbf{v}|\mathcal{X}; \mathcal{M}) \times \Pr[\{\mathbf{f}_s\}; \mathcal{K}_0]\right\}\right]. \tag{3.34}$$

More precisely, we give the details as follows.

Variational E-Step. For the variational distribution $q(\cdot)$ we use the *mean field approximation* (Wainwright and Jordan, 2008). That is, we assume a factorized distribution for disjoint groups of random variables. This results in an analytic tractable optimization problem. In addition, following Blei and Jordan (2006), the variational distribution approximates the distribution over $\mathbf{v}$ using a truncated stick-breaking representations, where for a fixed T, $q(v_T = 1) = 1$ and therefore $\pi_s(\mathbf{v}) = 0$, $s > T$. We fix the truncation level T while in general it can also be

treated as a variational parameter. Concretely, we propose the following factorized family of variational distributions over the hidden variables $\{\tilde{\mathbf{f}}, \mathbf{z}, \mathbf{v}\}$:

$$q(\tilde{\mathbf{f}}, \mathbf{z}, \mathbf{v}) = \prod_{s=1}^{T-1} q_s(v_s) \prod_{j=1}^{M} q_j(f_j, z_j). \tag{3.35}$$

Note that we do not assume any parametric form for $\{q_s, q_j\}$ and our only assumption is that the distribution factorizes into independent components. To optimize (3.33), recall the following result from (Bishop, 2006, Chapter 8):

Lemma 1 *Suppose we are given a probabilistic model with a joint distribution* $\Pr(\mathbf{X}, \mathbf{Z})$ *over* $\mathbf{X}, \mathbf{Z}$ *where* $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ *denote the observed variables and all the parameters and the hidden variables are given by* $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M\}$. *Assume the distribution of* $\mathbf{Z}$ *has the following form:*

$$q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(z_i).$$

Then, the ***KL*** *divergence between the posterior distribution* $\Pr(\mathbf{Z}|\mathbf{X})$ *and* $q(\mathbf{Z})$ *is minimized, and the optimal solution* $q_j^*(\mathbf{z}_j)$ *is given by*

$$q_j^*(\mathbf{z}_j) \propto \exp\left(\mathbb{E}_{i \neq j}\left[\log \Pr(\mathbf{X}, \mathbf{Z})\right]\right)$$

where $\mathbb{E}_{i \neq j}[\cdots]$ *denotes the expectation w.r.t.* $q()$ *over all* Z_i, $i \neq j$.

From the graphical model in Figure 3-3, the joint distribution of $\Pr(\mathcal{Y}, \tilde{\mathbf{f}}, \mathbf{z}, \mathbf{v}|\mathcal{X}; \mathcal{M}^g)$ can be written as:

$$\begin{aligned}\Pr(\mathcal{Y}, \tilde{\mathbf{f}}, \mathbf{z}, \mathbf{v}|\mathcal{X}) &= \Pr(\mathcal{Y}|\mathcal{X}, \tilde{\mathbf{f}}, \mathbf{z}) \Pr(\mathbf{z}|\mathbf{v}) \Pr(\tilde{\mathbf{f}}|\mathcal{X}) \Pr(\mathbf{v}|\alpha) \\ &= \prod_j \Pr(\boldsymbol{y}^j|\boldsymbol{x}^j, \tilde{\mathbf{f}}^j, z_j) \prod_j \Pr(z_j|\mathbf{v}) \prod_j \Pr(\tilde{\mathbf{f}}^j|\boldsymbol{x}^j) \prod_s \Pr(v_s|\alpha).\end{aligned}$$

Equivalently,

$$\begin{aligned}\log \Pr(\mathcal{Y}, \tilde{\mathbf{f}}, \mathbf{z}, \mathbf{v}|\mathcal{X}) = &\sum_j \log \Pr(\boldsymbol{y}^j|\boldsymbol{x}^j, \tilde{\mathbf{f}}^j, z_j) \\ &+ \sum_j \log \Pr(z_j|\mathbf{v}) + \sum_j \log \Pr(\tilde{\mathbf{f}}^j|\boldsymbol{x}^j) + \sum_s \log \Pr(v_s|\alpha).\end{aligned}$$

First we consider the distribution of $q_s(\mathbf{v})$. Following Blei and Jordan (2006), the second term can be expanded as

$$\log \Pr(z_j|\mathbf{v}) = \sum_{t=1}^{T} \mathbf{1}_{\{z_j>t\}} \log(1-v_t) + \mathbf{1}_{\{z_j=t\}} \log v_t \tag{3.36}$$

where $\mathbf{1}$ is the indicator function. Therefore, using the lemma above and denoting $\mathbf{v}\backslash v_s$ by $\mathbf{v}_{-s}$, and noting that the expectations of terms not including v_s are constant w.r.t. v_s, we have

$$\begin{aligned}
\log q_s(v_s) &\propto \mathbb{E}_{\tilde{\mathbf{f}},\mathbf{z},\mathbf{v}_{-s}} \left[\log \Pr(\mathcal{Y}, \tilde{\mathbf{f}}, \mathbf{z}, \mathbf{v}|\mathcal{X})\right] \\
&= \sum_j \left(\mathbb{E}_{\tilde{\mathbf{f}},\mathbf{z},\mathbf{v}_{-s}}[\mathbf{1}_{\{z_j>s\}}] \log(1-v_s) + \mathbb{E}_{\tilde{\mathbf{f}},\mathbf{z},\mathbf{v}_{-s}}[\mathbf{1}_{\{z_j=s\}}] \log v_s \right) \\
&\qquad + \log \Pr(v_s|\alpha) + \text{constant} \\
&= \sum_j \left(q(z_j > s) \log(1-v_s) + q(z_j = s) \log v_s \right) \\
&\qquad + \log \Pr(v_s|\alpha) + \text{constant}
\end{aligned}$$

Recalling that the prior is given by

$$v_s \sim \text{Beta}(1,\alpha) \propto v_s^{1-1}(1-v_s)^{\alpha-1},$$

we see that the distribution of $q_s(v_s)$ is

$$q_s(v_s) \propto v_s^{\sum_j q(z_j=s)} (1-v_s)^{\alpha + \sum_j \sum_{l=s+1}^{T} q(z_j=l) - 1}.$$

Observing the form of $q_s(v_s)$, we can see that it is a Beta distribution and $q_t(v_t) \sim \text{Beta}(\gamma_{t,1}, \gamma_{t,2})$ where

$$\begin{aligned}
\gamma_{t,1} &= 1 + \sum_j q(z_j = t) \\
\gamma_{t,2} &= \alpha + \sum_j \sum_{l=s+1}^{T} q(z_j = l).
\end{aligned}$$

We next consider $q_j(\tilde{\mathbf{f}}^j, z_j)$. Notice that we can always write $q_j(\tilde{\mathbf{f}}^j, z_j) = q_j(\tilde{\mathbf{f}}^j|z_j)q_j(z_j)$. Denote $h(z_j) = \mathbb{E}_{\mathbf{v}}\left[\log \Pr(z_j|\mathbf{v})\right]$, then again using the lemma above we have

$$\begin{aligned} q_j(\tilde{\mathbf{f}}^j|z_j)q_j(z_j) &\propto e^{h(z_j)} \Pr(\boldsymbol{y}^j|\boldsymbol{x}^j, \tilde{\mathbf{f}}^j, z_j) \Pr(\tilde{\mathbf{f}}^j|\boldsymbol{x}^j) \\ &= e^{h(z_j)} \Pr(\boldsymbol{y}^j, \tilde{\mathbf{f}}^j|\boldsymbol{x}^j, z_j) \\ &\propto e^{h(z_j)} \Pr(\boldsymbol{y}^j|\boldsymbol{x}^j, z_j) \Pr(\tilde{\mathbf{f}}^j|\boldsymbol{x}^j, \boldsymbol{y}^j, z_j) \\ &\propto \left[\underbrace{e^{h(z_j)} \Pr(\boldsymbol{y}^j|\boldsymbol{x}^j, z_j)}_{q_j(z_j)}\right] \left[\underbrace{\Pr(\tilde{\mathbf{f}}^j|\boldsymbol{x}^j, \boldsymbol{y}^j, z_j)}_{q_j(\tilde{\mathbf{f}}^j|z_j)}\right]. \end{aligned}$$

The equality in the second line holds because $\Pr(\tilde{\mathbf{f}}^j|\boldsymbol{x}^j) = \Pr(\tilde{\mathbf{f}}^j|\boldsymbol{x}^j, z_j)$; their distributions become coupled when conditioned on the observations $\boldsymbol{y}^j$, but without such observations they are independent. Therefore the left term yields

$$q_j(z_j) \propto e^{h(z_j)} \Pr(\boldsymbol{y}^j|\boldsymbol{x}^j, z_j)$$

where $\Pr(\boldsymbol{y}^j|\boldsymbol{x}^j, z_j)$ is given by (3.7). The value of $h(z_j)$ can be calculated using (3.36):

$$\begin{aligned} \log \Pr(z_j = s|\mathbf{v}) &= \sum_{t=1}^{s-1} \log(1 - v_t) + \log v_s \\ h(z_j = s) &= \mathbb{E}_{v_s}[\log v_s] + \sum_{i=1}^{s-1} \mathbb{E}_{v_i}\left[\log(1 - v_i)\right] \end{aligned}$$

where recalling that $q(v_t) \sim \text{Beta}(\gamma_{t,1}, \gamma_{t,2})$ and using properties of the Beta distribution, we have

$$\begin{aligned} \mathbb{E}_{v_t}[\log v_t] &= \Psi(\gamma_{i,1}) - \Psi(\gamma_{i,1} + \gamma_{i,2}) \\ \mathbb{E}_{v_i}\left[\log(1 - v_i)\right] &= \Psi(\gamma_{i,2}) - \Psi(\gamma_{i,1} + \gamma_{i,2}), \end{aligned} \tag{3.37}$$

where Ψ is the digamma function. Consequently, $q_j(z_j)$ has the following form

$$q_j(z_j = t) \propto \exp\left\{\mathbb{E}_{v_t}[\log v_t] + \sum_{i=1}^{t-1} \mathbb{E}_{v_i}\left[(1 - v_i)\right]\right\} \times \mathcal{N}\left(\mathbf{f}_j, \mathbf{K}_j^g + \sigma^2\mathbb{I}\right). \quad (3.38)$$

Note that this is the same form as in (3.6) of the previous model where $\Pr(z_j; \mathcal{M}^g)$ is replaced by $e^{h(z_j=t)}$.

Given z_j, $q_j(\tilde{\mathbf{f}}^j|z_j)$ is identical to (3.8) and leads to the conditional distribution such that

$$q_j(\tilde{\mathbf{f}}^j|z_j) \propto \Pr(\boldsymbol{y}^j|\boldsymbol{x}^j, \tilde{\mathbf{f}}^j, z_j) \Pr(\tilde{\mathbf{f}}^j; \boldsymbol{x}^j)$$

which is the posterior distribution under GP regression and thus is exactly the same form as in the previous model.

Variational M-Step. Denote $\widetilde{Q}$ as the expectation of the complete data log likelihood w.r.t. the hidden variables. Then as in (3.11), we have

$$\begin{aligned}
\widetilde{Q} &= \mathbb{E}_{q(\mathbf{v})}\mathbb{E}_{q(\mathbf{z})}\mathbb{E}_{q(\tilde{\mathbf{f}}|\mathbf{z})} \log\left(\prod_j \prod_s \left[\pi_s(\mathbf{v}) \Pr(\boldsymbol{y}^j|\tilde{\mathbf{f}}^j, z_j = s; \mathcal{M}) \Pr(\tilde{\mathbf{f}}^j; \mathcal{M})\right]^{z_{js}}\right) \\
&= \mathbb{E}_{\mathbf{v}}\left[\sum_{\mathbf{z}} q(\mathbf{z}) \left\{\sum_{j,s} z_{js} \cdot \int dq(\tilde{\mathbf{f}}^j|z_j = s) \right.\right. \\
&\qquad \left.\left. \times \log\left[\pi_s(\mathbf{v}) \Pr(\boldsymbol{y}^j|\tilde{\mathbf{f}}^j, z_j = s; \mathcal{M}) \Pr(\tilde{\mathbf{f}}^j; \mathcal{M})\right]\right\}\right] \\
&= \sum_{\mathbf{z}} q(\mathbf{z}) \left\{\sum_{j,s} z_{js} \cdot \int dq(\tilde{\mathbf{f}}^j|z_j = s) \right. \\
&\qquad \left. \times \log\left[\Pr(\boldsymbol{y}^j|\tilde{\mathbf{f}}^j, z_j = s; \mathcal{M}) \Pr(\tilde{\mathbf{f}}^j; \mathcal{M})\right]\right\} + \mathbb{E}_{\mathbf{v}}\left[\sum_{j,s} \log \pi_s(\mathbf{v})\right].
\end{aligned} \quad (3.39)$$

Notice that $\mathbb{E}_{\mathbf{v}}\left[\sum_{j,s} \log \pi_s(\mathbf{v})\right]$ is a constant w.r.t. the parameters of $\mathcal{M}$ and can be dropped in the optimization. Thus, following the same derivation as in the GMT

model, we have the form of the Q function as

$$\begin{aligned} Q(\mathcal{M}, \mathcal{M}^g) = &-\frac{1}{2}\sum_s \|\mathbf{f}_s\|^2_{\mathcal{H}_0} - \sum_j N_j \log\sigma \\ &- \frac{1}{2\sigma^2}\sum_{j,s}\gamma_{js}\mathbb{E}_{\{q(\tilde{\mathbf{f}}^j|z_j=s)\}}\left[\|\boldsymbol{y}^j - [\mathbf{f}_s * \delta_{t_j}](\boldsymbol{x}^j) - \tilde{\mathbf{f}}^j\|^2\right] \\ &+ \sum_{j,s}\gamma_{js}\mathbb{E}_{\{q(\tilde{\mathbf{f}}^j)\}}\left[\log\Pr(\tilde{\mathbf{f}}^j;\mathcal{M})\right]. \end{aligned} \tag{3.40}$$

where γ_{js} is given by (3.13). Now because the $q_j(z_j)$ and $q_j(f_j|z_j)$ have exactly the same form as before (except $\Pr(z_j;\mathcal{M}^g)$ is replaced by (3.38)), the previous derivation of the **M-Step** w.r.t. the parameter set $\mathcal{M}$ still holds.

To summarize, the algorithm is the same as Algorithm 1 except that

- we drop step 6,
- we add a step between steps 3 and 4 calculating $\gamma_{i,1}$ and $\gamma_{i,2}$ using (3.37),
- step 4 calculating (3.13) uses Equation (3.38) instead of (3.6).

3.4 Experiments

In this section, we evaluate our model and algorithms on a number of artificial and real datasets. The EM algorithm is restarted 5 times and the parameters that give the best Q value are selected. The EM algorithm stops when difference of the log-likelihood is less than 10e-5 or at a maximum of 200 iterations.

3.4.1 Regression on Synthetic data

In the first experiment, we demonstrate the performance of our algorithm on a regression task with artificial data. We generated the data following Assumption 2 under a mixture of three Gaussian processes. More precisely, each $f_s(x), s = 1, 2, 3$ is generated on the interval $[-50, 50]$ from a Gaussian process with covariance

function

$$\text{cov}[f_s(t_1), f_s(t_2)] = \exp\left\{-\frac{(t_1 - t_2)^2}{25}\right\}, \quad s = 1, 2, 3.$$

The individual effect $\tilde{f}_j$ is sampled via a Gaussian process with the covariance function

$$\text{cov}[\tilde{f}_j(t_1), \tilde{f}_j(t_2)] = 0.2 \exp\left\{-\frac{(t_1 - t_2)^2}{16}\right\}.$$

Then the hidden label z_j is sampled from a multinomial distribution with the parameter $\alpha = [1/3, 1/3, 1/3]$. The vector $\breve{\boldsymbol{x}}$ consists of 100 samples on $[-50, 50]$[1]. We fix a sample size N, each $\boldsymbol{x}^j$ includes N randomly chosen points from $\{\breve{x}_1, \cdots, \breve{x}_{100}\}$ and the observation $f^j(\boldsymbol{x}^j)$ is obtained as $(f_{z_j} + \tilde{f}_j)(\boldsymbol{x}^j)$. In the experiment, we vary the individual sample length N from 5 to 50. Finally, we generated 50 random tasks with the observation $\boldsymbol{y}^j$ for task j given by

$$\boldsymbol{y}^j \sim \mathcal{N}(f^j(\boldsymbol{x}^j), 0.01 \times \mathbb{I}), \quad j = 1, \cdots, 50.$$

The methods compared here include

1. **Single-task learning procedure (ST)**, where each $\mathbf{f}^j$ is estimated only using $\{\boldsymbol{x}_i^j, \boldsymbol{y}_i^j\}, i = 1, 2, \cdots, N$.

2. **Single center mixed-effects multi-task learning (SCMT)**, amounts to the mixed-effects model (Pillonetto et al., 2010) where one average function $\mathbf{f}$ is learned from $\{\boldsymbol{x}^j, \boldsymbol{y}^j\}, j = 1, \cdots, 50$ and $f^j = \mathbf{f} + \tilde{f}^j, j = 1, \cdots, 50$.

3. **Grouped mixed-effects model (GMT)**, the proposed method with number of clusters fixed to be the true model order ($k = 3$ in this case).

4. **Dirichlet process Grouped mixed-effects model (DP-GMT)**, the infinite mixture extension of the proposed model.

5. **"Cheating" grouped fixed-effect model (CGMT)**, which follows the same algorithm as the grouped mixed-effects model but uses the true label z_j in-

[1]The samples are generated via Matlab command: linspace(-50,50,100).

stead of their expectation for each task j. This serves as an upper bound for the performance of the proposed algorithm.

All algorithms (except for **ST** which does not estimate the kernel of the individual variations) use the same method to learn the kernel of the individual effects, which is assumed to be RBF, $\text{cov}[\tilde{f}_j(t_1), \tilde{f}_j(t_2)] = ae^{-\frac{(t_1-t_2)^2}{\sigma^2}}$. The Root Mean Square Error

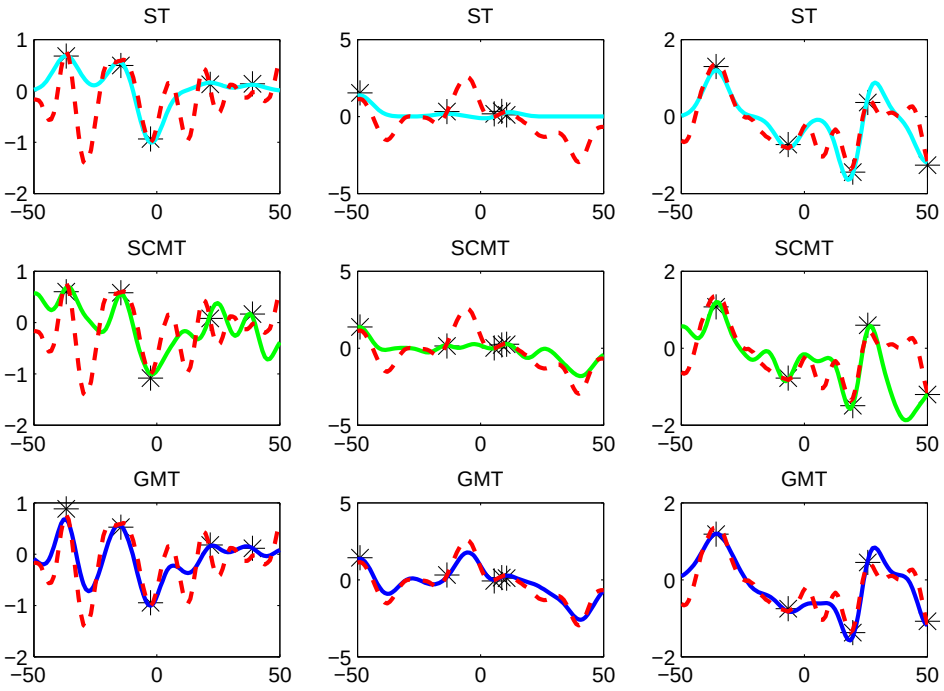

Figure 3-4: Simulated data: Comparison of the estimated function between single, multi-task and grouped multi-task. The red dotted line is the reference true function and the blue solid lines are the estimated functions.

(RMSE) for the four approaches is reported. For task j, the RMSE is defined as

$$\text{RMSE}_j = \sqrt{\frac{1}{100}\|f(\breve{\boldsymbol{x}}) - f^j(\breve{\boldsymbol{x}})\|^2}$$

where f is the learned function and RMSE for the data set is the mean of $\{\text{RMSE}_j\}, j = 1, \cdots, 50$. To illustrate the results qualitatively, we first plot in Figure 3-4 the true and learned functions in one trial. The left/center/right column illustrates one task that is sampled from group effect $\mathbf{f}_1$, $\mathbf{f}_2$ and $\mathbf{f}_3$. It it easy to see that, as ex-

pected, the tasks are poorly estimated under ST due the sparse sampling. The SCMT performs better than ST but its estimate is poor in areas where the three centers disagree. The estimates of GMT are much closer to the true function.

Figure 3-5 (Left) shows a comparison of the algorithms for 50 random data sets under the above setting when N equals 5. We see that GMT with the correct model order $k = 3$ almost always performs as well as its upper bound CGMT, illustrating that it recovers the correct membership of each task. On only three data sets, our algorithm is trapped in a local maximum yielding performance similar to SCMT and ST. Figure 3-5 (Right) shows the RMSE for increasing values of N for the same experimental setup. From the plot we can draw the conclusion that the proposed method works much better than SCMT and ST when the number of samples is fewer than 30. As the number of samples for each task increases, all methods are improving, but the proposed method always outperforms SCMT and ST in our experiments. Finally, all algorithms converge to almost the same performance level where the number of observations in each task are sufficient to recover the underlying function. Finally, Figure 3-5 also includes the performance of the DP-GMT on the same data. The truncation level of the Dirichlet process is 10 and the concentration parameter α is set to be 1. As we can see the DP-GMT is not distinguishable from the GMT (which has the correct k), indicating that model selection is successful in this example.

3.4.2 Classification on Astrophysics data

As we mentioned before, the concrete application motivating this research is the classification of stars into several meaningful categories from the astronomy literature. Classification is an important step within astrophysics research, as evidenced by published catalogs such as OGLE (Udalski et al., 1997) and MACHO (Alcock et al., 1993; Faccioli et al., 2007). However, the number of light sources in such surveys is increasing dramatically. For example Pan-STARRS (Hodapp et al., 2004) and LSST (Starr et al., 2002) collect data on the order of hundreds of billions of stars. Therefore, it is desirable to apply state-of-art machine learning techniques to

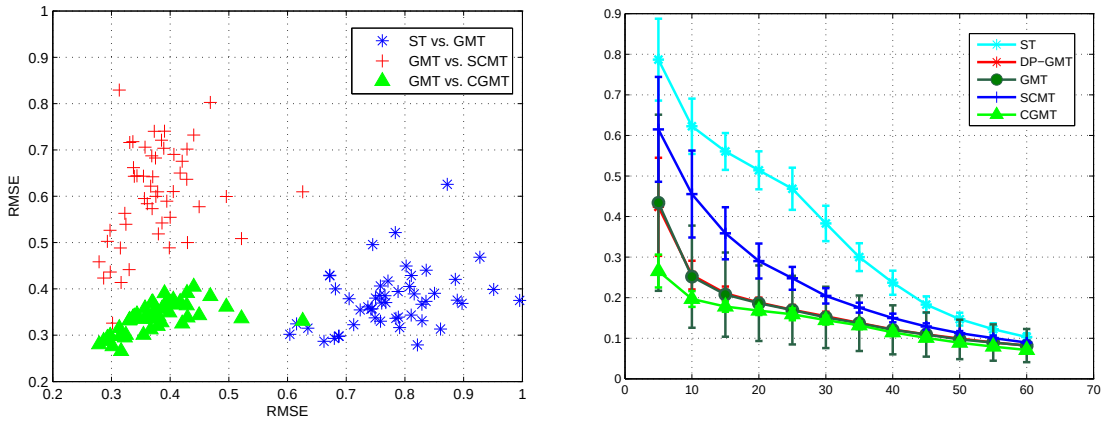

Figure 3-5: Left: Simulated data: Comparison between single task, multi-task, and grouped multi-task when sample size is 5. The figure gives 3 pairwise comparison. The Blue stars denote ST vs. GMT: we can see the GMT is better than ST since the stars are concentrated on the lower right. Similarly, the plot of red pluses demonstrates the advantage of GMT over SCMT and the plot of green triangles shows that the algorithm behaves almost as well as its upper bound. Right: Simulated data: Performance comparison of single, multi-task, CGMT, and DP grouped multi-task as a function of the number of samples per task.

enable automatic processing for astrophysics data classification.

The data from star surveys is normally represented by time series of brightness measurements, based on which they are classified into categories. Stars whose behavior is periodic are especially of interest in such studies. Figure 3-1 shows examples of such time series generated from the three major types of periodic variable stars: Cepheid, RR Lyrae, and Eclipsing Binary. In our experiments only stars of these classes are present in the data, and the period of each star is given.

We run our experiment on the OGLEII data set (Soszynski et al., 2003). This data set consists of 14087 time series from periodic variable stars with 3425 Cepheids, 3390 EBs and 7272 RRLs. We use the time series measurements in the I band (Soszynski et al., 2003). We perform several experiments with this data set to explore the potential of the proposed method. In previous work with this dataset Wachman et al. (2009) developed a kernel for periodic time series and used it with the support vector machine (SVM) (Cortes and Vapnik, 1995) to obtain good classification performance. We use the results of Wachman et al. (2009) as our baseline.[2]

[2]Wachman et al. (2009) used additional features, in addition to time series itself, to improve the

	UP + GMM	GMT	UP + 1-NN	K + SVM
RESULTS	0.956 ± 0.006	0.952 ± 0.005	0.865 ± 0.006	0.947 ± 0.005

Table 3.1: Accuracies with standard deviations reported on OGLEII dataset.

Classification using dense-sampled time series

In the first experiment, the time series are smoothed using a simple average filter, re-sampled to 50 points via linear-interpolation and normalized to have mean 0 and standard deviation of 1. Therefore, the time series are synchronously sampled in the pre-processing. We compare our method to Gaussian mixture model (GMM) and 1-Nearest Neighbor (1-NN). These two approaches are performed on the time series processed by Universal phasing (UP), which uses the method from Protopapas et al. (2006) to phase each time series to the sliding window on the time series with the maximum mean. We use a sliding window size of 5% of the number of original points; the phasing takes place after the pre-processing explained above. We learn a separate model for each class and for each class the model order for GMM and GMT is set to be 15.

We run 10-fold cross-validation (CV) over the entire data set and the results are shown in Table 3.1. We see that when the data is densely and synchronously sampled, the proposed method performs similar to the GMM, and they both outperform the kernel based results of Wachman et al. (2009). The similarity of the GMM and the proposed method under these experimental conditions is not surprising. The reason is that when the time series are synchronously sampled, aside from the difference of phasing, finding the group effect functions is reduced to estimating the mean vectors of the GMM. In addition, learning the kernel in the non-parametric approach is equivalent to estimating the covariance matrix of the GMM. More precisely, assuming that all time series are re-phased (that is, $t_j = 0$ for all j), the following results hold:

1. By placing a flat prior on the group effect function $\mathbf{f}_s, s = 1, \cdots, k$, or equivalently setting $\|\mathbf{f}_s\|^2_{\mathcal{H}_0} = 0$, Equation (3.18) is reduced to finding a vector $\mu_s \in \mathbb{N}$

classification performance. Here we focus on results using the time series only.

that minimizes $\sum_j \gamma_{js} \|\tilde{\mathbf{y}}^j - \mu_s\|^2$. Therefore, we obtain $\mathbf{f}_s = \mu_s = \sum_j \gamma_{js} \tilde{\mathbf{y}}_j / \sum_j \gamma_{js}$, which is exactly the mean of the sth cluster during the iteration of EM algorithm under the GMM setting.

2. The kernel $\mathbf{K}$ is learned in a non-parametric way. For the GP regression model, we see that considering noisy observations is essentially equivalent to considering non-noisy observations, but slightly modifying the model by adding a diagonal term on the covariance function for $\mathbf{f}_j$. Therefore, instead of estimating $\mathbf{K}$ and σ^2, it is convenient to put these two terms together, forming $\widehat{\mathbf{K}} = \mathbf{K} + \sigma^2 \mathbb{I}$. In other words, we add a σ^2 term to the variance of $\tilde{\mathbf{f}}^j$ and remove it from $\boldsymbol{y}^j$ which becomes deterministic. In this case, comparing to the derivation in Equation (3.7)—(3.10) we have $\tilde{\mathbf{f}}^j = \boldsymbol{y}^j - \mathbf{f}^j$ and $\tilde{\mathbf{f}}^j$ is determined given z_j. Comparing to Equation (3.8) we have the posterior mean $\boldsymbol{\mu}_{js}^g = \widehat{\mathbf{K}}\widehat{\mathbf{K}}^{-1}(\boldsymbol{y}^j - \mu_s) = \boldsymbol{y}^j - \mu_s$ and the posterior covariance matrix $\mathbf{C}_j^g$ vanishes. Applying these values in Equation (3.29) we get $\widehat{\mathbf{K}} = \frac{1}{M} \sum_j \sum_s \gamma_{js} (\boldsymbol{y}^j - \mu_s)(\boldsymbol{y}^j - \mu_s)^{\mathrm{T}}$. In the standard EM algorithm for the GMM, this is equal to the estimated covariance matrix when all k clusters are assumed to have the same variance.

Accordingly, when time series are synchronously sampled, the proposed model can be viewed as an extension of the Phased K-means (Rebbapragada et al., 2009). The Phased K-means (PKmeans) re-phases the time series before the similarity calculation and updates the centroids using the phased time series. Therefore, with shared covariance matrix, our model is a shift-invariant (Phased) GMM and the corresponding learning process is a Phased EM algorithm where each time series is re-phased in the **E** step. In experiments presented below we use Phased GMM directly in the feature space and generalize it so that each class has a separate covariance matrix.

We use the same experimental data to investigate the performance of the DP-GMT where the truncation level is set to be 30 and the concentration parameter α of the DP is set to be 1. The results are shown in Figure 3-6 and Table 3.2 where BIC-GMT means that the model order is chosen by BIC where the optimal k is chosen from 1 to 30. The poor performance of SCMT shows that a single center is

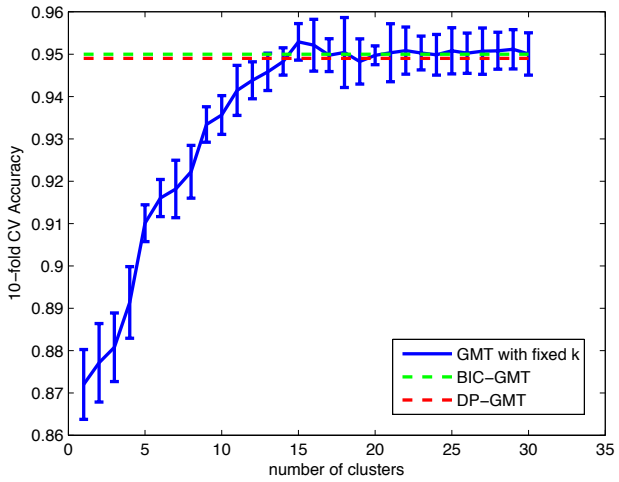

Figure 3-6: OGLEII data: Comparison of model selection methods using densely sampled data. The plot shows the performance of GMT with varying k, BIC for the GMT model, and DP-GMT. For visual clarity we only include the standard deviations on the GMT plot.

	SCMT	GMT	DP-GMT	BIC-GMT
RESULTS	0.874 ± 0.008	0.952 ± 0.005	0.949 ± 0.005	0.950 ± 0.002

Table 3.2: Accuracies with standard deviations reported on OGLEII dataset.

not sufficient for this data. As evident from the graph the DP-GMT is not distinguishable from the BIC-GMT. The advantage of the DP model is that this equivalent performance is achieved with much reduced computational cost because the BIC procedure must learn many models and choose among them whereas the DP learns a single model.

Classification using sparse-sampled time series

The OGLEII data set is in some sense a "nice" subset of the data from its corresponding star survey. Stars with small number of samples are often removed in pre-processing steps. For example, Wachman (2009) developed full system to process the MACHO survey and applied the kernel method to classify stars. In its pipeline, part of the preprocessing rejected 3.6 million light curves out of the approximately 25 million because of an insufficient number of observations. The proposed method potentially provides a way to include these instances in the clas-

sification process. In the second experiment, we demonstrate the performance of the proposed method on times series with sparse samples. Similar to the synthetic data, we started from sub-sampled versions of the original time series to simulate the condition that we would encounter in future star surveys.

Recall that our algorithm requires inverting $\mathcal{K}(\check{\boldsymbol{x}}, \check{\boldsymbol{x}})$ which is cubic in the number of sample points. For this dataset, $\check{\boldsymbol{x}}$ has more than 18000 measurements, making the inference infeasible. We therefore used a simple approximation in this experiment where we clip the samples to a fine grid of 200 equally spaced time points on $[0, 1]$, which is also the set of allowed time shifts. This avoids having a very high dimensional $\check{\boldsymbol{x}}$. While this is a crude method, it works reasonably well because the input x is single dimensional and we can use a fine grid for the clipping. In the next chapter, we develop a methodologically sound approach for such inference.

As in the previous experiment, each time series is universally phased, normalized and linearly-interpolated to length 50 to be plugged into GMM and 1-NN as well as the phased GMM mentioned above. The RBF kernel is used for the proposed method and we use model order 15 as above. Moreover, the performance for PKmeans is also presented, where the classification step is as follows: we learn the PKmeans model with $k = 15$ for each class and then the label of a new example is assigned to be the same as its closest centroid's label. PKmeans is also restarted 5 times and the best clustering is used for classification.

The results are shown in Figure 3-7 (Left). As can be easily observed, when each time series has sparse samples (i.e., number of samples per task is fewer than 30), the proposed method has a significant advantage over the other methods. As the number of samples per task increases, the proposed method improves quickly and performs close to its optimal performance given by previous experiment. Three additional aspects that call for discussion can be seen in the figure. First, note that for all three methods, the performance with dense data is lower than the results reported in Table 3.1. This can be explained by fact that the data set obtained by the interpolation of the sub-sampled measurements contains less information than that interpolated from the original measurements. Second, notice that the

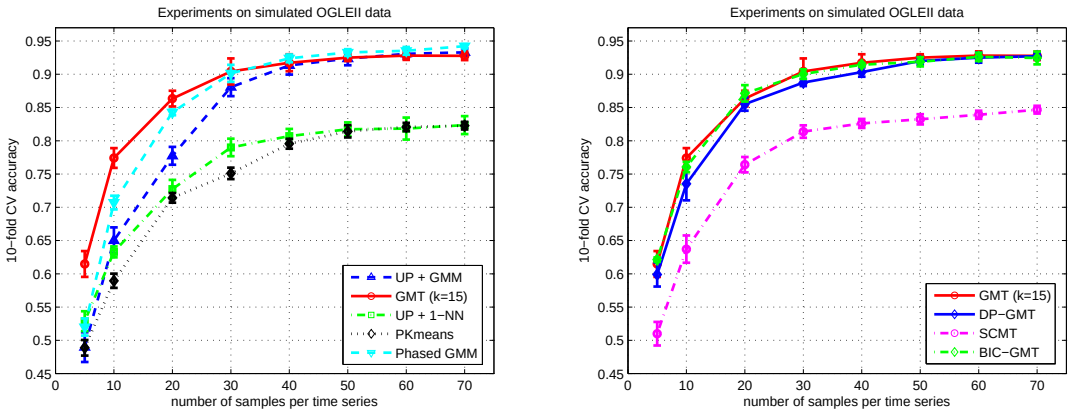

Figure 3-7: OGLEII data: Left: Comparison of algorithms with sparsely sampled data; Right: Comparison of model selection methods for GMT with sparsely sampled data.

Phased EM algorithm always outperforms the GMM plus UP demonstrating that re-phasing the time series inside the EM algorithm improves the results. Third, when the number of samples increases, the performance of the Phased EM gradually catches up and becomes better than the proposed method when each task has more than 50 samples. GMM plus universal phasing (UP) also achieves better performance when time series are densely sampled. One reason for the performance difference is the difference in the way the kernel is estimated. In Figure 3-7 GMT uses the parametric form of the kernel which is less expressive than getting precise estimates for every $\mathcal{K}(t_1, t_2)$. The GMM uses the non-parametric form which, given sufficient data, can lead to better estimates. A second reason can be attributed to the sharing of the covariance function in our model where the GMM and the Phased GMM do not apply this constraint.

Finally, we use the same experimental setting to compare the performance of various model selection models. The results are shown in Figure 3-7 (Right). The performance of BIC is not distinguishable from the optimal k selected in hindsight. The performance of DP is slightly lower but it comes close to these models.

To summarize, we conclude from the experiments with astronomy data that Phased EM is appropriate with densely sampled data but that the GMT and its variants should be used when data is sparsely and non-synchronously sampled.

In addition BIC coupled with GMT performs excellent model selection and DP does almost as well with a much reduced computational complexity.

Class discovery:

We show the potential of our model for class discovery by running the GMT model on the joint data set of the three classes (not using the labels). Then, each cluster is labeled according to the majority class of the instances that belong to the center. For a new test point, we determine which cluster it belongs to via the MAP probability and its label is given by the cluster to which it is assigned. We run 10 trials with different random initializations. In accordance with previous experiments that used 15 components per class we run GMT with model order of 45. We also run DP-GMT with a truncation level set to 90. The GMT obtains accuracy and standard deviation of $[0.895, 0.010]$ and the DP models obtains accuracy and standard deviation of $[0.925, 0.013]$. Note that it is hard to compare between the results because of the different model orders used. Rather than focus on the difference, the striking point is that we obtain almost pure clusters without using any label information. Given the size of the data set and the relatively small number of clusters this is a significant indication of the potential for class discovery in astrophysics.

3.5 Related Work

Classification of time series has attracted an increasing amount of interest in recent years due to its wide range of potential applications, for example ECG diagnosis (Wei and Keogh, 2006), EEG diagnosis (Lu et al., 2008), and Speech Recognition (Povinelli et al., 2004). Common methods choose some feature based representation or distance function for the time series (for example the sampled time points, or Fourier or wavelet coefficients as features and dynamic time warping for distance function) and then apply some existing classification method (Osowski et al., 2004; Ding et al., 2008). Our approach falls into another category, that is, model-based classification where the time series are assumed to be generated

by a probabilistic model and examples are classified using maximum likelihood or MAP estimates. A family of such models, closely related to the GMT, is discussed in detail below. Another common approach uses Hidden Markov models as a probabilistic model for sequence classification, and this has been applied to time series as well (Kim and Smyth, 2006).

Learning Gaussian processes from multiple tasks has previously been investigated in the hierarchical Bayesian framework, where a group of related tasks are assumed to share the same prior. Under this assumption, training points across all tasks are utilized to learn a better covariance function via the EM algorithm (Yu et al., 2005; Schwaighofer et al., 2005). In addition, Lu et al. (2008) extended the work of Schwaighofer et al. (2005) to a non-parametric mixed-effect model where each task can have its own random effect. Our model is based on the same algorithmic approach where the values of the function for each task at its corresponding points (i.e. $\{\tilde{\mathbf{f}}^j\}$ in our model) are considered as hidden variables. Furthermore, the proposed model is a natural generalization of Schwaighofer et al. (2005) where the fixed-effect function is sampled from a mixture of regression functions each of which is a realization of a common Gaussian process. Along a different dimension, our model differs from the infinite mixtures of Gaussian processes model for clustering (Jackson et al., 2007) in two aspects: first, instead of using zero mean Gaussian process, we allow the mean functions to be sampled from another Gaussian process; second, the individual variation in our model serves as the covariance function in their model but all mixture components share the same kernel.

Although having a similar name, the Gaussian process *mixture of experts* model focuses mainly on the issues of non-stationarity in regression (Rasmussen and Ghahramani, 2002; Tresp, 2001). By dividing the input space into several (even infinite) regions via a gating network, the Gaussian process mixture of expert model allows different Gaussian processes to make predictions for different regions.

In terms of the clustering aspect, our work is most closely related to the so-called *mixture of regressions* (Gaffney and Smyth, 2005, 2003; Gaffney, 2004; Gaffney and Smyth, 1999). The name comes from the fact that these approaches substi-

tute component density models with conditional regression density models in the framework of standard mixture model. For phased time series, Gaffney and Smyth (1999) first proposed the regression-based mixture model where they used Polynomial and Kernel regression models for the mean curves. Further, Gaffney and Smyth (2003) integrated the linear random effects models with mixtures of regression functions. In their model, each time series is sampled by a parametric regression model whose parameters are generated from a Gaussian distribution. To incorporate the time shifts, Chudova et al. (2003) proposed a shift-invariant Gaussian mixture model for multidimensional time series. They constrained the covariance matrices to be diagonal to handle the non-synchronous case. They also treated time shifts as hidden variables and derived the EM algorithm under full Bayesian settings, i.e. where each parameter has a prior distribution. Furthermore, Gaffney and Smyth (2005) developed a generative model for misaligned curves in a more general setting. Their joint clustering-alignment model also assumes a normal parametric regression model for the cluster labels, and Gaussian priors on the hidden transformation variables which consist of shifting and scaling in both the time and magnitude. Our model extends the work of Gaffney and Smyth (2003) to admit non-parametric Bayesian regression mixture models and at the same time handle the non-phased time series. If the group effects are assumed to have a flat prior, our model differs from Chudova et al. (2003) in the following two aspects in addition to the difference of Bayesian treatment. First, our model does not include the time shifts as hidden variables but instead estimates them as parameters. Second, we can handle shared full covariance matrix instead of diagonal ones by using a parametric form of the kernel. On the other hand, given the time grid $\check{x}$, we can design the kernel for individual variations as $\mathcal{K}(\check{x}_i, \check{x}_j) = a_i \delta_{ij}(\check{x}_i, \check{x}_j), i, j = 1, \cdots, \mathbb{N}$. Using this choice, our model is the same as Chudova et al. (2003) with shared diagonal covariance matrix. In summary, in additional to being non-parametric and thus more flexible, our model allows a more flexible structure of the covariance matrix that can treat synchronized and non-synchronized time series in a unified framework, but at the same time it is

constrained to have the same covariance matrix across all clusters.

3.6 Conclusion

We developed a novel Bayesian nonparametric multi-task learning model (GMT) where each task is modeled as a sum of a group-specific function and an individual task function with a Gaussian process prior. We also extended the model such that the number of groups is not bounded using a Dirichlet process mixture model (DP-GMT). We derive efficient EM and variational EM algorithms to learn the parameters of the models and demonstrated their effectiveness using experiments in regression, classification and class discovery. Our models are particularly useful for sparsely and non-synchronously sampled time series data, and model selection can be effectively performed with these models.

Chapter 4

Sparse Grouped Mixed-effects GP

One of the main difficulties with the GMT model and algorithms in the previous chapter, is computational cost. While the number of samples per task (N_j) is small, the total sample size $\sum_j N_j$ can be huge, and the cubic complexity of GP inference can be prohibitively large. Some improvement can be obtained when all the input tasks share the same sampling points, or when different tasks share many of the input points (Pillonetto et al., 2009, 2010). However, if the number of distinct sampling points is large the complexity remains high. In particular, this is the case in some of the experiments of the previous chapter where sample points are clipped to a fine grid to avoid the high cardinality of the example set.

The same problem, handling large samples, has been addressed in single task formalizations of GP, where several approaches for so-called sparse solutions have been developed (Rasmussen and Williams, 2006; Seeger and Lawrence, 2003; Snelson and Ghahramani, 2006; Titsias, 2009). These methods approximate the GP with $m \ll N$ support variables (or inducing variables, pseudo inputs) $\mathcal{X}_m$ and their corresponding function values $\mathbf{f}_m$ and perform inference using this set. Thus, instead of clipping to a grid, new sampling points are adaptively chosen and their values are estimated instead of being "moved to" nearby data points.

In this chapter, we develop a methodologically sound sparse solution for the grouped mixed-effect GP model. Specifically, we extend the approach of Titsias (2009) and develop a variational approximation that allows us to efficiently learn

the shared hyper-parameters and choose the sparse pseudo samples. In addition, we show how the variational approximation can be used to perform prediction efficiently once learning has been performed. Our approach is particularly useful when individual tasks have a small number of samples, different tasks do not share sampling points, and there is a large number of tasks. Our experiments, using artificial and real data, validate the approach showing that it can recover the performance of inference with the full sample, that it performs better than simple sparse approaches for multi-task GP, and that for some applications it significantly outperforms alternative sparse multi-task GP formulations (Álvarez and Lawrence, 2011).

The rest of this chapter is organized as follows. Section 4.1 reviews the mixed-effect GP model and its direct inference. Section 4.2 develops the variational inference and model selection for the sparse mixed-effect GP model. Section 4.3 shows how to extend the sparse solution to the grouped mixed-effect GP model. We discuss related work in Section 4.5 and demonstrate the performance of the proposed approach using three datasets in Section 4.4. The last section concludes with a summary and directions for future work.

4.1 Mixed-effects GP for Multi-task Learning

In the next section, we develop the mixed-effect model and its sparse solution without considering grouping, i.e., using Assumption 1. The model and results are extended to include grouping in Section 4.3.

To prepare for these derivations, we start by reviewing direct inference in the mixed-effect model. Assumption 1 implies that for $j, l \in \{1, \cdots, M\}$, the following holds:

$$\mathbf{Cov}[f^j(\boldsymbol{s}), f^l(\boldsymbol{t})] = \mathbf{Cov}[\bar{f}(\boldsymbol{s}), \bar{f}(\boldsymbol{t})] + \delta_{jl} \cdot \mathbf{Cov}[\tilde{f}(\boldsymbol{s}), \tilde{f}(\boldsymbol{t})] \tag{4.1}$$

where δ_{jl} is the Kronecker delta function. Given data $\mathcal{D}^j = \{(\boldsymbol{x}_i^j, y_i^j)\}, i = 1, 2, \cdots, N_j$

and let $\mathcal{X}$ be the concatenation of the examples from all tasks $\mathcal{X} = (\boldsymbol{x}_i^j)$, and similarly let $\mathcal{Y} = (\boldsymbol{y}_i^j)$, where $i = 1, 2, \cdots, N_j, j = 1, 2, \cdots, M$ and $N = \sum_j N_j$. It can easily been seen that, for any $j \in \{1, \cdots, M\}$ and new input $\boldsymbol{x}^*$ for task j, we have

$$\begin{bmatrix} f^j(\boldsymbol{x}^*) \\ \mathcal{Y} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}^{\dagger}(\boldsymbol{x}^*, \boldsymbol{x}^*) & \mathbf{K}^{\dagger}(\boldsymbol{x}^*, \mathcal{X}) \\ \mathbf{K}^{\dagger}(\mathcal{X}, \boldsymbol{x}^*) & \mathbf{K}^{\dagger}(\mathcal{X}, \mathcal{X}) + \sigma^2 \mathbb{I} \end{bmatrix}\right) \tag{4.2}$$

where the covariance matrix $\mathbf{K}^{\dagger}$ is given by

$$\mathbf{K}^{\dagger}((\boldsymbol{x}_i^j), (\boldsymbol{x}_k^l)) = \mathcal{K}(\boldsymbol{x}_i^j, \boldsymbol{x}_k^l) + \delta_{jl} \cdot \widetilde{\mathcal{K}}((\boldsymbol{x}_i^j, \boldsymbol{x}_k^l).$$

From (4.2) we can extract the marginal distribution $\Pr(\mathcal{Y})$ where

$$\mathcal{Y}|\mathcal{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}^{\dagger}(\mathcal{X}, \mathcal{X}) + \sigma^2 \mathbb{I}), \tag{4.3}$$

which can be used for model selection, that is, learning the hyper-parameters of the GP. Equation (4.2) also provides the predictive distribution where

$$\begin{aligned} \mathbb{E}(f^j(\boldsymbol{x}^*)|\mathcal{Y}) &= \mathbf{K}^{\dagger}(\boldsymbol{x}^*, \mathcal{X})(\mathbf{K}^{\dagger}(\mathcal{X}, \mathcal{X}) + \sigma^2 \mathbb{I})^{-1} \mathcal{Y} \\ \mathbf{Var}(f^j(\boldsymbol{x}^*)|\mathcal{Y}) &= \mathbf{K}^{\dagger}(\boldsymbol{x}^*, \boldsymbol{x}^*) - \mathbf{K}^{\dagger}(\boldsymbol{x}^*, \mathcal{X})(\mathbf{K}^{\dagger}(\mathcal{X}, \mathcal{X}) + \sigma^2 \mathbb{I})^{-1} \mathbf{K}^{\dagger}(\mathcal{X}, \boldsymbol{x}^*). \end{aligned} \tag{4.4}$$

This works well in that sharing the information improves predictive performance but, as the number of tasks grows, the dimension N increases leading to slow inference scaling as $\mathcal{O}(N^3)$. In other words, even though each task may have a very small sample, the multi-task inference problem becomes infeasible when the number of tasks is large.

In single task GP regression, to reduce the computational cost, several sparse GP approaches have been proposed (Rasmussen and Williams, 2006; Seeger and Lawrence, 2003; Snelson and Ghahramani, 2006; Titsias, 2009). In general, these methods approximate the GP with a small number $m \ll N$ of support variables and perform inference using this subset and the corresponding function values $\mathbf{f}_m$. Different approaches differ in how they choose the support variables and the sim-

plest approach is to choose a random subset of the given data points. Recently, Titsias (2009) introduced a sparse method based on variational inference using a set $\mathcal{X}_m$ of inducing samples, which are different from the training points. In this approach, the sample points $\mathcal{X}_m$ are chosen to maximize a variational lower bound on the marginal likelihood, therefore providing a clear methodology for the choice of the support set. Following their idea, Álvarez et al. (2010) proposed the variational inference for sparse convolved multiple output GPs.

In this chapter we extend this approach to provide a sparse solution for the aforementioned model as well as generalizing it to the Grouped mixed-effect GP model. As in the case of sparse methods for single task GP, the key idea is to introduce a small set of m auxiliary inducing sample points $\mathcal{X}_m$ and base the learning and inference on these points. For the multi-task case, each $\tilde{f}^j(\cdot)$ is specific to the jth task. Therefore, it makes sense to induce values only for the fixed-effect portion $\mathbf{f}_m = \bar{f}(\mathcal{X}_m)$. The details of this construction are developed in the following sections.

4.2 Sparse Mixed-effects GP Model

In this section, we develop a sparse solution for the mixed-effects model without group effect. The model is simpler to analyze and apply, and it thus provides a good introduction to the results developed in the next section for the grouped model.

4.2.1 Variational Inference

In this section we specify the sparse model, and show how we can learn the hyperparameters and the inducing variables using the sparse model. As mentioned above, we introduce auxiliary inducing sample points $\mathcal{X}_m$ and hidden variables $\mathbf{f}_m = \bar{f}(\mathcal{X}_m)$. Let $\mathbf{f}^j = \bar{f}(\boldsymbol{x}^j) \in \mathbb{R}^{N_j}$ and $\tilde{\mathbf{f}}^j = \tilde{f}(\boldsymbol{x}^j) \in \mathbb{R}^{N_j}$ denote the values of the two functions at $\boldsymbol{x}^j$. In addition let $\mathbf{K}_{*j} = \mathbf{K}(\boldsymbol{x}^*, \boldsymbol{x}^j)$, $\mathbf{K}_{jj} = \mathbf{K}(\boldsymbol{x}^j, \boldsymbol{x}^j)$ and $\mathbf{K}_{mm} = \mathbf{K}(\mathcal{X}_m, \mathcal{X}_m)$, and similarly for $\widetilde{\mathbf{K}}_{*j}, \widetilde{\mathbf{K}}_{jj}, \widetilde{\mathbf{K}}_{mm}$. We wish to maximize the

marginal likelihood $\Pr(\mathcal{Y})$ and perform the inference (i.e., calculate the posterior distribution over the hidden variables).

Recall the variational EM method introduced in Section 3.3.3. We are given observed data $\mathbf{X}$, hidden variables $\mathbf{Z}$, and we introduce the distribution $q(\mathbf{Z})$ to approximate $\Pr(\mathbf{Z}|\mathbf{X})$. The variational lower bound of the log likelihood is given by

$$\ln \Pr(\mathbf{X}|\boldsymbol{\theta}) = \ln \int q(\mathbf{Z}) \left[\frac{\Pr(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} \right] \mathrm{d}\mathbf{Z} \geqslant \int q(\mathbf{Z}) \ln \left[\frac{\Pr(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} \right] \mathrm{d}\mathbf{Z},$$

where the inequality holds because of Jensen's inequality. In this section we use exactly the same lower bound as in (3.32), but this time through an observation due to Titsias (2009), we are able to optimize it directly and do not need the EM algorithm.

In the following, we evaluate and optimize the variational lower bound of the proposed sparse model. To this end, we need the complete data likelihood and the variational distribution. The complete data likelihood is given by:

$$\begin{aligned}
&\Pr(\{\boldsymbol{y}^j\}, \{\mathbf{f}^j, \tilde{\mathbf{f}}^j\}, \mathbf{f}_m) \\
&= \Pr(\{\boldsymbol{y}^j\}|\{\mathbf{f}^j, \tilde{\mathbf{f}}^j\}) \Pr(\{\tilde{\mathbf{f}}^j\}) \Pr(\{\mathbf{f}^j\}|\mathbf{f}_m) \Pr(\mathbf{f}_m) \\
&= \left[\prod_{j=1}^{M} \Pr(\boldsymbol{y}^j|\mathbf{f}^j, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j) \right] \Pr(\{\mathbf{f}^j\}|\mathbf{f}_m) \Pr(\mathbf{f}_m).
\end{aligned}$$

Then, we approximate the posterior $\Pr(\{\mathbf{f}^j, \tilde{\mathbf{f}}^j\}, \mathbf{f}_m|\{\boldsymbol{y}^j\})$ on the hidden variables by

$$q(\{\mathbf{f}^j, \tilde{\mathbf{f}}^j\}, \mathbf{f}_m) = \left[\prod_{j=1}^{M} \Pr(\tilde{\mathbf{f}}^j|\mathbf{f}^j, \boldsymbol{y}^j) \right] \Pr(\{\mathbf{f}^j\}|\mathbf{f}_m) \phi(\mathbf{f}_m) \tag{4.5}$$

which extends the variational form used by Titsias (2009) to handle the individual variations as well as the multiple tasks. One can see that the variational distribution is not completely in free form. Instead, $q(\cdot)$ preserves the exact form of $\Pr(\tilde{\mathbf{f}}^j|\mathbf{f}^j, \boldsymbol{y}^j)$ and in using $\Pr(\{\mathbf{f}^j\}|\mathbf{f}_m)$ it implicitly assumes that $\mathbf{f}_m$ is a sufficient statistic for $\{\mathbf{f}^j\}$. The free form $\phi(\mathbf{f}_m)$ corresponds to $\Pr(\mathbf{f}_m|\mathcal{Y})$ but allows it to di-

verge from this value to compensate for the assumption that $\mathbf{f}_m$ is sufficient. Notice that we are not making any assumption about the sufficiency of $\mathbf{f}_m$ in the generative model and the approximation is entirely due to the variational distribution. An additional assumption is added later to derive a simplified form of the predictive distribution.

With the two ingredients ready, the variational lower bound (Jordan et al., 1999; Bishop, 2006), denoted as $F_V(\mathcal{X}_m, \phi)$, is given by:

$$\begin{aligned}
\log \Pr(\mathcal{Y}) &\geqslant F_V(\mathcal{X}_m, \phi) \\
&= \int q(\{\mathbf{f}^j, \tilde{\mathbf{f}}^j\}, \mathbf{f}_m) \times \log \left[\frac{\Pr(\{\boldsymbol{y}^j\}, \{\mathbf{f}^j, \tilde{\mathbf{f}}^j\}, \mathbf{f}_m)}{q(\{\mathbf{f}^j, \tilde{\mathbf{f}}^j\}, \mathbf{f}_m)} \right] d\{\mathbf{f}^j\} d\{\tilde{\mathbf{f}}^j\} d\mathbf{f}_m \\
&= \int \left[\prod_{j=1}^{M} \Pr(\tilde{\mathbf{f}}^j | \mathbf{f}^j, \boldsymbol{y}^j) \right] \Pr(\{\mathbf{f}^j\} | \mathbf{f}_m) \phi(\mathbf{f}_m) \\
&\quad \times \log \left[\prod_{l=1}^{M} \frac{\Pr(\boldsymbol{y}^l | \mathbf{f}^l, \tilde{\mathbf{f}}^l) \Pr(\tilde{\mathbf{f}}^l)}{\Pr(\tilde{\mathbf{f}}^l | \mathbf{f}^l, \boldsymbol{y}^l)} \cdot \frac{\Pr(\mathbf{f}_m)}{\phi(\mathbf{f}_m)} \right] d\{\mathbf{f}^j\} d\{\tilde{\mathbf{f}}^j\} d\mathbf{f}_m \\
&= \int \phi(\mathbf{f}_m) \left\{ \log G(\mathbf{f}_m, \mathcal{Y}) + \log \left[\frac{\Pr(\mathbf{f}_m)}{\phi(\mathbf{f}_m)} \right] \right\} d\mathbf{f}_m.
\end{aligned}$$

The inner integral denoted as $\log G(\mathbf{f}_m, \mathcal{Y})$ is

$$\begin{aligned}
&\int \left[\prod_{j=1}^{M} \Pr(\tilde{\mathbf{f}}^j | \mathbf{f}^j, \boldsymbol{y}^j) \right] \Pr(\{\mathbf{f}^j\} | \mathbf{f}_m) \times \sum_{l=1}^{M} \log \left[\frac{\Pr(\boldsymbol{y}^l | \mathbf{f}^l, \tilde{\mathbf{f}}^l) \Pr(\tilde{\mathbf{f}}^l)}{\Pr(\tilde{\mathbf{f}}^l | \mathbf{f}^l, \boldsymbol{y}^l)} \right] d\{\mathbf{f}^j\} d\{\tilde{\mathbf{f}}^j\} \\
&= \sum_{j=1}^{M} \int \Pr(\tilde{\mathbf{f}}^j | \mathbf{f}^j, \boldsymbol{y}^j) \Pr(\mathbf{f}^j | \mathbf{f}_m) \times \log \left[\frac{\Pr(\boldsymbol{y}^j | \mathbf{f}^j, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j | \mathbf{f}^j, \boldsymbol{y}^j)} \right] d\mathbf{f}^j d\tilde{\mathbf{f}}^j
\end{aligned} \tag{4.6}$$

where the second line holds because in the sum indexed by l all the product measures

$$\prod_{j=1, j \neq l}^{M} \Pr(\tilde{\mathbf{f}}^j | \mathbf{f}^j, \boldsymbol{y}^j) \Pr(\{\mathbf{f}^n\}_{n \neq l} | \mathbf{f}_m, \mathbf{f}_l) d\{\mathbf{f}^j\} d\{\tilde{\mathbf{f}}^j\},$$

are integrated to 1, leaving only the j-th integral. In the following we show that

$$\log G(\mathbf{f}_m, \mathcal{Y}) = \sum_{j=1}^{m} \left[\log \left[\mathcal{N}(\boldsymbol{y}^j | \boldsymbol{\alpha}_j, \widehat{\mathbf{K}}_{jj}) \right] - \frac{1}{2} \mathbf{Tr} \left[(\mathbf{K}_{jj} - \mathbf{Q}_{jj}) [\widehat{\mathbf{K}}_{jj}]^{-1} \right] \right] \tag{4.7}$$

where $\boldsymbol{\alpha}_j = \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{f}_m$, $\widehat{\mathbf{K}}_{jj} = \sigma_j^2\mathbb{I} + \widetilde{\mathbf{K}}_{jj}$, and $\mathbf{Q}_{jj} = \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mj}$. Thus we have

$$\begin{aligned} F_V(\mathcal{X}_m, \phi) &= \int \phi(\mathbf{f}_m) \left[\log G(\mathbf{f}_m, \mathcal{Y}) + \log\left[\frac{\Pr(\mathbf{f}_m)}{\phi(\mathbf{f}_m)}\right]\right] d\mathbf{f}_m \\ &= \int \phi(\mathbf{f}_m) \log\left[\frac{\prod_j \left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j, \widehat{\mathbf{K}}_{jj})\right] \Pr(\mathbf{f}_m)}{\phi(\mathbf{f}_m)}\right] d\mathbf{f}_m \\ &\quad - \frac{1}{2}\sum_{j=1}^{M} \mathbf{Tr}\left[(\mathbf{K}_{jj} - \mathbf{Q}_{jj})[\widehat{\mathbf{K}}_{jj}]^{-1}\right]. \end{aligned} \tag{4.8}$$

Let $\boldsymbol{v}$ be a random variable and g any function, then by Jensen's inequality $\mathbb{E}[\log g(\boldsymbol{v})] \leq \log \mathbb{E}[g(\boldsymbol{v})]$. Therefore, the best lower bound we can derive from (4.8), if it is achievable, is the case where equality holds in Jensen's inequality. It follows that $\phi(\mathbf{f}_m)$ can be chosen to obtain equality, and therefore, the variational lower bound is

$$F_V(\mathcal{X}_m, \phi) = \log \int \prod_j \left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j, \widehat{\mathbf{K}}_{jj})\right] \Pr(\mathbf{f}_m) d\mathbf{f}_m - \frac{1}{2}\sum_{j=1}^{M} \mathrm{Tr}\left[(\mathbf{K}_{jj} - Q_{jj})[\widehat{\mathbf{K}}_{jj}]^{-1}\right].$$

Evaluating the integral by marginalizing out $\mathbf{f}_m$ and recalling that $\mathcal{Y}$ is the concatenation of the $\boldsymbol{y}^j$, we get

$$F_V(\mathcal{X}_m, \boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) = \log\left[\mathcal{N}(\mathcal{Y}|\mathbf{0}, \boldsymbol{\Lambda}_m\mathbf{K}_{mm}^{-1}\boldsymbol{\Lambda}_m^T + \widehat{\mathbf{K}}^m)\right] - \sum_{j=1}^{M}\left[\frac{1}{2}\mathbf{Tr}\left[(\mathbf{K}_{jj} - \mathbf{Q}_{jj})[\widehat{\mathbf{K}}_{jj}]^{-1}\right]\right] \tag{4.9}$$

where

$$\boldsymbol{\Lambda}_m = \begin{pmatrix} \mathbf{K}_{1m} \\ \mathbf{K}_{2m} \\ \vdots \\ \mathbf{K}_{Mm} \end{pmatrix} \quad \text{and} \quad \widehat{\mathbf{K}}^m = \bigoplus_{j=1}^{M} \widehat{\mathbf{K}}_{jj} = \begin{pmatrix} \widehat{\mathbf{K}}_{11} & & & \\ & \widehat{\mathbf{K}}_{22} & & \\ & & \ddots & \\ & & & \widehat{\mathbf{K}}_{MM} \end{pmatrix}.$$

In (4.9), we have explicitly written the parameters that can be chosen to further optimize the lower bound, namely the support inputs $\mathcal{X}_m$, and the hyper-parameters

$\boldsymbol{\theta}$ and $\tilde{\boldsymbol{\theta}}$ in $\mathcal{K}$ and $\widetilde{\mathcal{K}}$ respectively. Finally, observe that in (4.9), F_V does not depend on $q(\cdot)$ and we are able to obtain a closed form for $\phi^*(\mathbf{f}_m)$ directly. As a result, we can optimize F_V with respect to both $q(\cdot)$ and the parameters directly and do not need to resort to the EM algorithm. In contrast, in general, the variational lower bound normally contains both $q(\cdot)$ and other parameters, making the EM algorithm necessary, as in Section 3.3.3 and later in Section 4.3.

By calculating derivatives of (4.9) we can optimize the lower bound using a gradient based method. We provide the details of the gradients and their computation in the next Section where we discuss the solution of the grouped model. In the experiments in this Chapter, we use stochastic gradient descent (SGD), which works better than the Conjugate Gradient (CG) in this scenario where the number of tasks is large, by using a heuristic search over subsets of points in the dataset. As in the previous chapter, this is reasonable because the input time is single dimensional, but gradients can be hard to use in the high dimensional case or with discrete inputs. Titsias (2009) outlines methods that can be used when gradients are not useful.

Evaluating $\log G(\mathbf{f}_m, \mathcal{Y})$

Consider the j-th element in the sum of (4.6):

$$
\begin{aligned}
\widehat{G}_j(\mathbf{f}^j, \boldsymbol{y}^j) &= \int \Pr(\tilde{\mathbf{f}}^j|\mathbf{f}^j, \boldsymbol{y}^j) \Pr(\mathbf{f}^j|\mathbf{f}_m) \log \left[\frac{\Pr(\boldsymbol{y}^j|\mathbf{f}^j, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j|\mathbf{f}^j, \boldsymbol{y}^j)} \right] d\mathbf{f}^j d\tilde{\mathbf{f}}^j \\
&= \int \Pr(\tilde{\mathbf{f}}^j|\mathbf{f}^j, \boldsymbol{y}^j) \Pr(\mathbf{f}^j|\mathbf{f}_m) \\
&\quad \times \log \left[\frac{\Pr(\tilde{\mathbf{f}}^j|\boldsymbol{y}^j, \mathbf{f}^j) \Pr(\boldsymbol{y}^j|\mathbf{f}^j)}{\Pr(\tilde{\mathbf{f}}^j|\mathbf{f}^j)} \cdot \frac{\Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j|\mathbf{f}^j, \boldsymbol{y}^j)} \right] d\mathbf{f}^j d\tilde{\mathbf{f}}^j \\
&= \int \Pr(\mathbf{f}^j|\mathbf{f}_m) \log \left[\Pr(\boldsymbol{y}^j|\mathbf{f}^j) \right] \left(\int \Pr(\tilde{\mathbf{f}}^j|\mathbf{f}^j, \boldsymbol{y}^j) d\tilde{\mathbf{f}}^j \right) d\mathbf{f}^j \\
&= \int \Pr(\mathbf{f}^j|\mathbf{f}_m) \log \left[\Pr(\boldsymbol{y}^j|\mathbf{f}^j) \right] d\mathbf{f}^j = \mathbb{E}_{[\mathbf{f}^j|\mathbf{f}_m]} \log \left[\Pr(\boldsymbol{y}^j|\mathbf{f}^j) \right]
\end{aligned}
$$

where the third line holds because of the independence between $\tilde{\mathbf{f}}^j$ and $\mathbf{f}^j$. We next show how this expectation can be evaluated. This is more complex than the single-

task case (Titsias, 2009) because of the coupling of the fixed-effect and the random effect. Recall that

$$\Pr(\mathbf{f}^j|\mathbf{f}_m) = \mathcal{N}(\mathbf{f}^j|\mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{f}_m, \mathbf{K}_{jj} - \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mj})$$

and

$$\boldsymbol{y}^j|\mathbf{f}^j \sim \mathcal{N}(\mathbf{f}^j, \widehat{\mathbf{K}}_{jj})$$

where $\widehat{\mathbf{K}}_{jj} = \sigma^2\mathbb{I} + \widetilde{\mathbf{K}}_{jj}$. Denote $\widehat{\mathbf{K}}_{jj}^{-1} = \mathbf{L}^{\mathrm{T}}\mathbf{L}$ where $\mathbf{L}$ can be chosen as its Cholesky decomposition. We have

$$\begin{aligned}\log\left[\Pr(\boldsymbol{y}^j|\mathbf{f}^j)\right] &= -\frac{1}{2}(\boldsymbol{y}^j - \mathbf{f}^j)^T\widehat{\mathbf{K}}_{jj}^{-1}(\boldsymbol{y}^j - \mathbf{f}^j) + \log\left[(2\pi)^{-\frac{N_j}{2}}\right] + \log\left[|\widehat{\mathbf{K}}_{jj}|^{-\frac{1}{2}}\right] \\ &= -\frac{1}{2}(\mathbf{L}\boldsymbol{y}^j - \mathbf{L}\mathbf{f}^j)^T(\mathbf{L}\boldsymbol{y}^j - \mathbf{L}\mathbf{f}^j) + \log\left[(2\pi)^{-\frac{N_j}{2}}\right] + \log\left[|\widehat{\mathbf{K}}_{jj}|^{-\frac{1}{2}}\right].\end{aligned}$$

Notice that

$$\Pr(\mathbf{L}\mathbf{f}^j|\mathbf{f}_m) = \mathcal{N}(\mathbf{L}\mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{f}_m, \mathbf{L}(\mathbf{K}_{jj} - \mathbf{Q}_{jj})\mathbf{L}^{\mathrm{T}})$$

where $\mathbf{Q}_{jj} = \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mj}$. Recall the fact that for $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and a constant vector $\boldsymbol{a}$, we have $\mathbb{E}[\|\boldsymbol{a} - \boldsymbol{x}\|^2] = \|\boldsymbol{a} - \boldsymbol{\mu}\|^2 + \mathbf{Tr}(\boldsymbol{\Sigma})$. Thus,

$$\begin{aligned}\mathbb{E}_{[\mathbf{f}^j|\mathbf{f}_m]}\log\left[\Pr(\boldsymbol{y}^j|\mathbf{f}^j)\right] &= -\frac{1}{2}\|\mathbf{L}\boldsymbol{y}^j - \mathbf{L}\mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{f}_m\|^2 \\ &\quad - \frac{1}{2}\mathbf{Tr}(\mathbf{L}(\mathbf{K}_{jj} - \mathbf{Q}_{jj})\mathbf{L}^T) + \log\left[(2\pi)^{-\frac{N_j}{2}}\right] + \log\left[|\widehat{\mathbf{K}}_{jj}|^{-\frac{1}{2}}\right] \\ &= \left\{ -\frac{1}{2}\left[\boldsymbol{y} - \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{f}_m\right]^T\left(L^TL\right)\left[\boldsymbol{y} - \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{f}_m\right]\right. \\ &\quad \left. + \log\left[(2\pi)^{-\frac{N_j}{2}}\right] + \log\left[|\mathbf{K}_{jj}|^{-\frac{1}{2}}\right]\right\} - \frac{1}{2}\mathrm{Tr}\left[L(\mathbf{K}_{jj} - \mathbf{Q}_{jj})L^T\right] \\ &= \log\left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j, \widehat{\mathbf{K}}_{jj})\right] - \frac{1}{2}\mathbf{Tr}\left[(\mathbf{K}_{jj} - \mathbf{Q}_{jj})\widehat{\mathbf{K}}_{jj}^{-1}\right]\end{aligned}$$

where $\boldsymbol{\alpha}_j = \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{f}_m$. Finally, calculating $\sum_j \widehat{G}_j(\mathbf{f}^j, \boldsymbol{y}^j)$ we get (4.7).

Variational distribution $\phi^*(\mathbf{f}_m)$

For equality to hold in Jensen's inequality, the function inside the log must be constant. In our case this is easily achieved because $\phi(\mathbf{f}_m)$ is a free parameter, and we can set

$$\left[\frac{\prod_j \left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j, \widehat{\mathbf{K}}_{jj})\right] \Pr(\mathbf{f}_m)}{\phi(\mathbf{f}_m)}\right] \equiv c,$$

yielding the bound given in (4.9). Setting $\phi(\mathbf{f}_m) \propto \prod_j \left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j, \widehat{\mathbf{K}}_{jj})\right] \Pr(\mathbf{f}_m)$ yields the form of the optimal variational distribution

$$\begin{aligned}\phi^*(\mathbf{f}_m) &\propto \prod_j \left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j, \widehat{\mathbf{K}}_{jj})\right] \Pr(\mathbf{f}_m) \\ &\propto \exp\left\{ -\frac{1}{2}\mathbf{f}_m^T \left[\mathbf{K}_{mm}^{-1}\boldsymbol{\Phi}\mathbf{K}_{mm}^{-1}\right] \mathbf{f}_m + \mathbf{f}_m^{\mathrm{T}} \left(\mathbf{K}_{mm}^{-1} \sum_j \mathbf{K}_{mj} \left[\widehat{\mathbf{K}}_{jj}\right]^{-1} \boldsymbol{y}^j\right)\right\},\end{aligned}$$

from which we observe that $\phi^*(\mathbf{f}_m)$ is

$$\mathcal{N}\left(\mathbf{f}_m \middle| \mathbf{K}_{mm}\boldsymbol{\Phi}^{-1} \sum_j \mathbf{K}_{mj}[\widehat{\mathbf{K}}_{jj}]^{-1}\boldsymbol{y}_j, \mathbf{K}_{mm}\boldsymbol{\Phi}^{-1}\mathbf{K}_{mm}\right) \tag{4.10}$$

where $\boldsymbol{\Phi} = \mathbf{K}_{mm} + \sum_j \mathbf{K}_{mj}[\widehat{\mathbf{K}}_{jj}]^{-1}\mathbf{K}_{jm}$. Notice that by choosing the number of tasks to be 1 and the random effect to be a noise process, i.e. $\widetilde{K}(s,t) = \sigma^2\delta(s,t)$, (4.9) and (4.10) are exactly the variational lower bound and the corresponding variational distribution in (Titsias, 2009).

4.2.2 Prediction using the Variational Solution

Given any task j, our goal is to calculate the predictive distribution of $f^j(\boldsymbol{x}^*) = \bar{f}(\boldsymbol{x}^*) + \tilde{f}^j(\boldsymbol{x}^*)$ at some new input point $\boldsymbol{x}^*$. As described before, the full inference is expensive and therefore we wish to use the variational approximation for the predictions as well. The key assumption is that $\mathbf{f}_m$ contains as much information as $\mathcal{Y}$ in terms of making prediction for $\bar{f}$. This will be made explicit below. To start with, it is easy to see that the predictive distribution is Gaussian and that it satisfies

$$\begin{aligned}\mathbb{E}[f^j(\boldsymbol{x}^*)|\mathcal{Y}] &= \mathbb{E}[\bar{f}(\boldsymbol{x}^*)|\mathcal{Y}] + \mathbb{E}[\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{Y}] \\ \mathbf{Var}[f^j(\boldsymbol{x}^*)|\mathcal{Y}] &= \mathbf{Var}[\bar{f}(\boldsymbol{x}^*)|\mathcal{Y}] + \mathbf{Var}[\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{Y}] + 2\mathbf{Cov}[\bar{f}(\boldsymbol{x}^*), \tilde{f}^j(\boldsymbol{x}^*)|\mathcal{Y}].\end{aligned} \tag{4.11}$$

The above equation is more complex than the predictive distribution for single-task sparse GP (Titsias, 2009) because of the coupling induced by $\bar{f}(\boldsymbol{x}^*), \tilde{f}^j(\boldsymbol{x}^*)|\mathcal{Y}$. We next show how this can be calculated via conditioning.

To calculate the terms in (4.11), three parts are needed, i.e., $\Pr(\bar{f}(\boldsymbol{x}^*)|\mathcal{Y})$, $\Pr(\tilde{f}(\boldsymbol{x}^*)|\mathcal{Y})$ and $\mathbf{Cov}[\bar{f}(\boldsymbol{x}^*), \tilde{f}_j(\boldsymbol{x}^*)|\mathcal{Y}]$. Using the assumption of the variational form given in (4.5), we have the following facts,

1. $\mathbf{f}_m|\mathcal{Y} \sim \phi^*(\mathbf{f}_m) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{A})$ where $\boldsymbol{\mu}$ and $\mathbf{A}$ are given in (4.10).

2. $\mathbf{f}_m$ is sufficient for $\{\mathbf{f}^j\}$, i.e. $\Pr(\{\mathbf{f}^j\}|\mathbf{f}_m, \mathcal{Y}) = \Pr(\{\mathbf{f}^j\}|\mathbf{f}_m)$. Since we are interested in prediction for each task separately, by marginalizing out $\mathbf{f}^l, l \neq j$, we also have $\Pr(\mathbf{f}^j|\mathbf{f}_m, \mathcal{Y}) = \Pr(\mathbf{f}^j|\mathbf{f}_m)$ and

$$\mathbf{f}^j|\mathbf{f}_m, \mathcal{Y} \sim \mathcal{N}\left(\mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{f}_m, \mathbf{K}_{jj} - \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mj}\right). \tag{4.12}$$

3. For $\tilde{f}^j(\boldsymbol{x}^*)$ we can view $\boldsymbol{y}^j - \mathbf{f}^j$ as noisy realizations from the same GP as $\tilde{f}^j(\boldsymbol{x}^j)$ and therefore

$$\tilde{f}^j(\boldsymbol{x}^*)|\mathbf{f}^j, \mathcal{Y} \sim \mathcal{N}\left(\widetilde{\mathbf{K}}_{*j}\left[\widetilde{\mathbf{K}}_{jj} + \sigma_j^2\mathbb{I}_j\right]^{-1}\left[\boldsymbol{y}^j - \mathbf{f}^j\right], \widetilde{\mathbf{K}}_{**} - \widetilde{\mathbf{K}}_{*j}\left[\widetilde{\mathbf{K}}_{jj} + \sigma_j^2\mathbb{I}_j\right]^{-1}\widetilde{\mathbf{K}}_{j*}\right). \tag{4.13}$$

In order to obtain a sparse form of the predictive distribution we need to make an additional assumption.

Assumption 4 *We assume that* $\mathbf{f}_m$ *is sufficient for* $\bar{f}(\boldsymbol{x}^*)$*, i.e.,*

$$\Pr(\bar{f}(\boldsymbol{x}^*)|\mathbf{f}_m, \mathcal{Y}) = \Pr(\bar{f}(\boldsymbol{x}^*)|\mathbf{f}_m),$$

implying that

$$\bar{f}(\boldsymbol{x}^*)|\mathbf{f}_m, \mathcal{Y} \sim \mathcal{N}\left(\mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{f}_m, \mathbf{K}_{**} - \mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{K}_{m*}\right). \tag{4.14}$$

The above set of conditional distributions also imply that $\bar{f}(\boldsymbol{x}^*)$ and $\tilde{f}^j(\boldsymbol{x}^*)$ are independent given $\mathbf{f}_m$ and $\mathcal{Y}$.

To evaluate (4.11), we have the following:

Firstly, we can easily get $\Pr(\bar{f}(\boldsymbol{x}^*)|\mathcal{Y})$ by marginalizing out $\mathbf{f}_m|\mathcal{Y}$ in (4.14),

$$\Pr(\bar{f}(\boldsymbol{x}^*)|\mathcal{Y}) = \int \Pr(\bar{f}(\boldsymbol{x}^*)|\mathbf{f}_m)\phi^*(\mathbf{f}_m)d\mathbf{f}_m$$

yielding

$$\bar{f}(\boldsymbol{x}^*)|\mathcal{Y} \sim \mathcal{N}\left(\mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mu, \mathbf{K}_{**} - \mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{K}_{m*} + \mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}A\mathbf{K}_{mm}^{-1}\mathbf{K}_{m*}\right). \tag{4.15}$$

Similarly, we can obtain $\Pr(\tilde{f}(\boldsymbol{x}^*)|\mathcal{Y})$ by first calculating $\Pr(\mathbf{f}^j|\mathcal{Y})$ by marginalizing out $\mathbf{f}_m|\mathcal{Y}$ in (4.12) and then marginalizing out $\mathbf{f}^j|\mathcal{Y}$ in (4.13), as follows. First we have $\mathbf{f}^j|\mathcal{Y} \sim \mathcal{N}(\mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\boldsymbol{\mu}, \mathbf{B})$ where

$$\mathbf{B} = \mathbf{K}_{jj} - \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mj} + \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}A\mathbf{K}_{mm}^{-1}\mathbf{K}_{mj}.$$

Next for $\Pr(\tilde{f}(\boldsymbol{x}^*)|\mathcal{Y})$, we have

$$\Pr(\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{Y}) = \int \Pr(\tilde{f}^j(\boldsymbol{x}^*)|\mathbf{f}^j, \boldsymbol{y}^j)\Pr(\mathbf{f}^j|\mathcal{Y})d\mathbf{f}^j$$

and marginalizing out $\mathbf{f}^j$, $\tilde{f}(\boldsymbol{x}^*)|\mathcal{Y}$ can be obtained as

$$\begin{aligned}\mathcal{N}\Bigg(&\widetilde{\mathbf{K}}_{*j}\left[\widetilde{\mathbf{K}}_{jj} + \sigma_j^2\mathbb{I}_j\right]^{-1}\left(\boldsymbol{y}^j - \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mu\right), \widetilde{\mathbf{K}}_{**} - \widetilde{\mathbf{K}}_{*j}\left[\widetilde{\mathbf{K}}_{jj} + \sigma_j^2\mathbb{I}_j\right]^{-1}\widetilde{\mathbf{K}}_{j*} \\ &+ \widetilde{\mathbf{K}}_{*j}\left[\widetilde{\mathbf{K}}_{jj} + \sigma_j^2\mathbb{I}_j\right]^{-1}\mathbf{K}_{jm}\mathbf{K}_{mm}^{-1} \times \mathbf{B} \times \mathbf{K}_{mm}^{-1}\mathbf{K}_{mj}\left(\widetilde{\mathbf{K}}_{jj} + \sigma_j^2\mathbb{I}_j\right)^{-1}\widetilde{\mathbf{K}}_{j*}\Bigg).\end{aligned} \tag{4.16}$$

Finally, to calculate $\mathbf{Cov}[\bar{f}(\boldsymbol{x}^*)\tilde{f}^j(\boldsymbol{x}^*)|\mathcal{Y}]$ we have

$$\mathbf{Cov}[\bar{f}(\boldsymbol{x}^*), \tilde{f}_j(\boldsymbol{x}^*)|\mathcal{Y}] = \mathbb{E}\left[\bar{f}^j(\boldsymbol{x}^*) \cdot \tilde{f}^j(\boldsymbol{x}^*)|\mathcal{Y}\right] - \mathbb{E}\left[\bar{f}(\boldsymbol{x}^*)|\mathcal{Y}\right]\mathbb{E}\left[\tilde{f}(\boldsymbol{x}^*)|\mathcal{Y}\right]$$

where

$$\begin{aligned}\mathbb{E}\left[\bar{f}^j(\boldsymbol{x}^*) \cdot \tilde{f}^j(\boldsymbol{x}^*)|\mathcal{Y}\right] &= \mathbb{E}_{\mathbf{f}_m|\mathcal{Y}}\mathbb{E}\left[\bar{f}^j(\boldsymbol{x}^*) \cdot \tilde{f}^j(\boldsymbol{x}^*)|\mathbf{f}_m, \mathcal{Y}\right] \\ &= \mathbb{E}_{\mathbf{f}_m|\mathcal{Y}}\left[\mathbb{E}\left[\bar{f}^j(\boldsymbol{x}^*)|\mathbf{f}_m\right] \cdot \mathbb{E}\left[\tilde{f}^j(\boldsymbol{x}^*)|\mathbf{f}_m, \boldsymbol{y}^j\right]\right]\end{aligned} \tag{4.17}$$

where the second line holds because, as observed above, the terms are conditionally independent. The first term $\mathbb{E}\left[\bar{f}^j(\boldsymbol{x}^*)|\mathbf{f}_m\right]$ can be obtained directly from (4.14). By marginalizing out $\mathbf{f}^j|\mathbf{f}_m$ in (4.13) such that

$$\Pr(\tilde{f}^j(\boldsymbol{x}^*)|\mathbf{f}_m, \boldsymbol{y}^j) = \int \Pr(\tilde{f}^j(\boldsymbol{x}^*)|\mathbf{f}^j, \mathcal{Y}) \Pr(\mathbf{f}^j|\mathbf{f}_m) d\mathbf{f}^j,$$

we can get the second term. This yields

$$\begin{aligned}\mathcal{N}\Bigg(&\widetilde{\mathbf{K}}_{*j}\left[\widetilde{\mathbf{K}}_{jj} + \sigma_j^2\mathbb{I}_j\right]^{-1}\left(\boldsymbol{y}^j - \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{f}_m\right), \widetilde{\mathbf{K}}_{**} - \widetilde{\mathbf{K}}_{*j}\left[\widetilde{\mathbf{K}}_{jj} + \sigma_j^2\mathbb{I}_j\right]^{-1}\widetilde{\mathbf{K}}_{j*} \\ &+ \widetilde{\mathbf{K}}_{*j}\left[\widetilde{\mathbf{K}}_{jj} + \sigma_j^2\mathbb{I}_j\right]^{-1}\mathbf{C}\left(\widetilde{\mathbf{K}}_{jj} + \sigma_j^2\mathbb{I}_j\right)^{-1}\widetilde{\mathbf{K}}_{j*}\Bigg)\end{aligned} \tag{4.18}$$

where $\mathbf{C} = \mathbf{K}_{jj} - \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mj}$. To simplify the notation, let $\mathbf{H} = \mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}$, $\mathbf{F} = \widetilde{\mathbf{K}}_{*j}\left(\widetilde{\mathbf{K}}_{jj} + \sigma_j^2\mathbb{I}_j\right)^{-1}$ and $\mathbf{G} = \mathbf{K}_{jm}\mathbf{K}_{mm}^{-1}$. Then (4.17) can be evaluated as

$$\mathbf{H}\boldsymbol{y}^j\mathbf{F} \cdot \mathbb{E}[\mathbf{f}_m] - \mathbf{F}\mathbf{G}\left(\mathbb{E}\left[\mathbf{f}_m\mathbf{f}_m^{\mathrm{T}}|\mathcal{Y}\right]\right)\mathbf{H}^{\mathrm{T}} = \mathbf{H}\boldsymbol{y}^j\mathbf{F} \cdot \boldsymbol{\mu} - \mathbf{F}\mathbf{G}\left[\mathbf{A} + \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}}\right]\mathbf{H}^{\mathrm{T}}.$$

We have therefore shown how to calculate the predictive distribution in (4.11). The complexity of these computations is $\mathcal{O}(N_j^3 + m^3)$ which is a significant improvement over $\mathcal{O}(N^3)$ where $N = M \times N_j$.

4.3 Sparse Grouped Mixed-effects GP Model

In this section, we extend the sparse solution of the mixed-effect GP model to the grouped mixed-effect model. We abuse notation and still call it GMT. We show how to perform the inference and model selection efficiently.

4.3.1 Generative Model

First, we specify the sparse GMT (SGMT) model, and show how we can learn the hyper-parameters and the inducing variables using this sparse model. The generative process (shown in Fig. 4-1) is as follows, where **Dir** and **Multi** denote the Dirichlet and the Multinominal distribution respectively.

1. Draw the mean effect functions: $\bar{f}_k(\cdot)|\boldsymbol{\theta}_k \sim \mathcal{GP}(0, \mathcal{K}_k(\cdot,\cdot)), \quad k = 1, 2, \cdots, K$;

2. Draw $\boldsymbol{\pi}|\boldsymbol{\alpha}_0 \sim \mathbf{Dir}(\boldsymbol{\alpha}_0)$;

3. For the j-th task (time series);

   - Draw $z_j|\boldsymbol{\pi} \sim \mathbf{Multi}(\boldsymbol{\pi})$;
   - Draw the random effect: $\tilde{f}^j(\cdot)|\tilde{\boldsymbol{\theta}} \sim \mathcal{GP}(0, \widetilde{\mathcal{K}}(\cdot,\cdot))$;
   - Draw $\boldsymbol{y}^j|z_j, f^j, \boldsymbol{x}^j, \sigma_j^2 \sim \mathcal{N}\left(f^j(\boldsymbol{x}^j), \sigma_j^2 \cdot \mathbb{I}_j\right)$, where $f^j = \bar{f}_{z_j} + \tilde{f}^j$ and where to simplify the notation $\mathbb{I}_j$ stands for $\mathbb{I}_{N_j}$.

This is slightly different from the model of Chapter 3 in that we no longer model the phase shift aspect and in that we provide prior over $\boldsymbol{\pi}$ and no longer treat it as a parameter. In contrast with Chapter 3 and following Section 4.2, our algorithm below also treats $\bar{f}$ as a hidden variable and does not estimate it directly as as parameter. Finally, in contrast with Chapter 3, we do learn the kernel of the fixed-effects and each center uses its own kernel function $\mathcal{K}_k$.

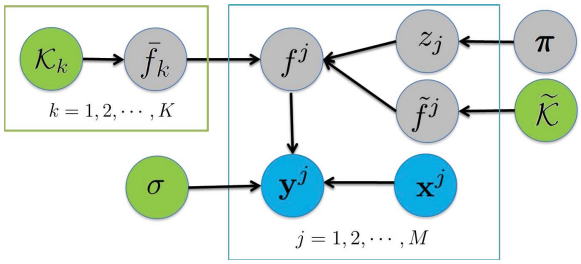

Figure 4-1: Plate graph of the GMT-GP. Blue nodes denote observations, green nodes are (hyper)parameters and the gray nodes are latent variables.

4.3.2 Variational Inference

In this section we show how to perform the learning via variational approximation. The derivation follows the same outline as in the previous section but due to the hidden variables z_j that specify group membership, we have to use the variational EM algorithm. As mentioned above, for the k-th mixed-effect (or center), we introduce m_k auxiliary inducing support variables $\mathcal{X}_m^k$ and the hidden variable $\boldsymbol{\eta}_k = \bar{f}_k(\mathcal{X}_m^k)$, which is the value of k-th fixed-effect function evaluated at $\mathcal{X}_m^k$.

Let $\mathfrak{f}_k = \bar{f}_k(\mathcal{X}) \in \mathbb{R}^N$ denote the function values of the k-th mean effect so that $\mathbf{f}_k^j = \bar{f}_k(\boldsymbol{x}^j) \in \mathbb{R}^{N_j}$ is the sub-vector of $\mathfrak{f}_k$ corresponding to the j-th task. Let $\tilde{\mathbf{f}}^j = \tilde{f}(\boldsymbol{x}^j) \in \mathbb{R}^{N_j}$ be the values of the random effect at $\boldsymbol{x}^j$. Denote the collection of the hidden variables as $\mathfrak{F} = \{\mathfrak{f}_k\}, \widetilde{\mathcal{F}} = \{\tilde{\mathbf{f}}^j\}, \mathbf{H} = \{\boldsymbol{\eta}_k\}, \mathbf{Z} = \{z_j\}$, and $\boldsymbol{\pi}$. In addition let $\mathbf{K}_{*j}^k = \mathcal{K}_k(\boldsymbol{x}^*, \boldsymbol{x}^j)$, $\mathbf{K}_{jj}^k = \mathcal{K}_k(\boldsymbol{x}^j, \boldsymbol{x}^j), \mathbf{K}_{jk} = \mathcal{K}_k(\boldsymbol{x}^j, \mathcal{X}_m^k)$ and $\mathbf{K}_{kk} = \mathcal{K}_k(\mathcal{X}_m^k, \mathcal{X}_m^k)$, and similarly $\widetilde{\mathbf{K}}_{*j} = \widetilde{\mathcal{K}}(\boldsymbol{x}^*, \boldsymbol{x}^j), \widetilde{\mathbf{K}}_{jj} = \widetilde{\mathcal{K}}(\boldsymbol{x}^j, \boldsymbol{x}^j)$ and $\widehat{\mathbf{K}}_{jj} = \widetilde{\mathbf{K}}_{jj} + \sigma_j^2 \mathbb{I}_j$ where $\mathbb{I}_j$ stands for $\mathbb{I}_{N_j}$.

To learn the hyper-parameters we wish to maximize the marginal likelihood $\Pr(\mathcal{Y})$. In the following we develop a variational lower bound for this quantity. To this end, we need the complete data likelihood and the variational distribution. The complete data likelihood is given by

$$\Pr(\mathcal{Y}, \mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi}) = \Pr(\mathcal{Y}|\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{Z}) \Pr(\mathfrak{F}|\mathbf{H}) \Pr(\mathbf{Z}|\boldsymbol{\pi}) \Pr(\boldsymbol{\pi}) \Pr(\widetilde{\mathcal{F}}) \Pr(\mathbf{H}) \quad (4.19)$$

where

$$\Pr(\mathbf{H}) = \prod_{k=1}^{K} \Pr(\boldsymbol{\eta}_k), \qquad \Pr(\widetilde{\mathcal{F}}) = \prod_{j=1}^{M} \Pr(\tilde{\mathbf{f}}^j),$$

$$\Pr(\boldsymbol{\pi}) = \mathbf{Dir}(\boldsymbol{\pi}|\alpha_0), \qquad \Pr(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{j=1}^{M}\prod_{k=1}^{K} \pi_k^{z_{jk}}$$

$$\Pr(\mathfrak{F}|\mathbf{H}) = \prod_{k=1}^{K} \Pr(\mathfrak{f}_k|\boldsymbol{\eta}_k), \quad \Pr(\mathcal{Y}|\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{Z}) = \prod_{j=1}^{M}\prod_{k=1}^{K} \left[\Pr(\boldsymbol{y}^j|\tilde{\mathbf{f}}^j, \mathfrak{f}_k)\right]^{z_{jk}}$$

where, as usual $\{z_{jk}\}$ represent z_j as a unit vector. We approximate the posterior $\Pr(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi}|\mathcal{Y})$ on the hidden variables using

$$q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi}) = q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}|\mathbf{Z})q(\mathbf{Z})q(\boldsymbol{\pi}) \tag{4.20}$$

where

$$\begin{aligned} q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}|\mathbf{Z}) &= \Pr(\widetilde{\mathcal{F}}|\mathfrak{F}, \mathbf{Z}, \mathcal{Y}) \Pr(\mathfrak{F}|\mathbf{H})\boldsymbol{\Phi}(\mathbf{H}) \\ &= \prod_{j=1}^{M}\prod_{k=1}^{K} \left[\Pr(\tilde{\mathbf{f}}^j|\mathfrak{f}_k, \boldsymbol{y}^j)\right]^{z_{jk}} \prod_{k=1}^{K} \Pr(\mathfrak{f}_k|\boldsymbol{\eta}_k)\phi(\boldsymbol{\eta}_k). \end{aligned}$$

This extends the variational form of the previous section to handle the grouping. Our use of $\mathfrak{f}_k$ as the complete set of observation when the true group is k makes for a convenient notation and simplifies the derivation. The variational lower bound,

denoted as F_V, is given by:

$$
\begin{aligned}
&\log \Pr(\mathcal{Y}) \geqslant F_V \\
&\quad \int q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi}) \times \log \left[\frac{\Pr(\mathcal{Y}, \mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi})}{q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H}, \mathbf{Z}, \boldsymbol{\pi})} \right] d\mathfrak{F}\, d\widetilde{\mathcal{F}}\, d\mathbf{H}\, d\mathbf{Z}\, d\boldsymbol{\pi} \\
&\quad = \int q(\boldsymbol{\pi}) q(\mathbf{Z}) q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H} | \mathbf{Z}) \\
&\qquad \times \log \left[\frac{\Pr(\mathcal{Y} | \mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{Z}) \Pr(\mathfrak{F} | \mathbf{H}) \Pr(\mathbf{Z} | \boldsymbol{\pi}) \Pr(\pi) \Pr(\widetilde{\mathcal{F}}) \Pr(\mathbf{H})}{q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H} | \mathbf{Z}) q(\mathbf{Z}) q(\boldsymbol{\pi})} \right] d\mathfrak{F} d\widetilde{\mathcal{F}} d\mathbf{H} d\mathbf{Z} d\boldsymbol{\pi} \\
&\quad = \int q(\mathbf{Z}) q(\boldsymbol{\pi}) \log \left[\frac{\Pr(\boldsymbol{\pi}) \Pr(\mathbf{Z} | \boldsymbol{\pi})}{q(\mathbf{Z}) q(\boldsymbol{\pi})} \right] d\boldsymbol{\pi} d\mathbf{Z} \\
&\qquad + \int q(\mathbf{Z}) q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H} | \mathbf{Z}) \log \left[\frac{\Pr(\mathcal{Y} | \mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{Z}) \Pr(\mathfrak{F} | \mathbf{H}) \Pr(\widetilde{\mathcal{F}}) \Pr(\mathbf{H})}{q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H} | \mathbf{Z})} \right] d\mathfrak{F} d\widetilde{\mathcal{F}} d\mathbf{H} d\mathbf{Z}
\end{aligned}
$$

To begin with, we evaluate the second term denoted as F_{V2}, as follows. The term inside the log can be evaluated as

$$
\begin{aligned}
\boldsymbol{\Delta} &= \frac{\prod_{j,k} \left[\Pr(\boldsymbol{y}^j | \tilde{\mathbf{f}}^j, \mathfrak{f}_k) \right]^{z_{jk}} \prod_k \Pr(\mathfrak{f}_k | \boldsymbol{\eta}_k) \prod_j \Pr(\tilde{\mathbf{f}}^j) \prod_k \Pr(\boldsymbol{\eta}_k)}{\prod_{j,k} \left[\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j) \right]^{z_{jk}} \prod_k \Pr(\mathfrak{f}_k | \boldsymbol{\eta}_k) \phi(\boldsymbol{\eta}_k)} \\
&= \prod_{j=1}^{m} \prod_{k=1}^{K} \left[\frac{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j)} \right]^{z_{jk}} \times \prod_{k=1}^{K} \frac{\Pr(\boldsymbol{\eta}_k)}{\phi(\boldsymbol{\eta}_k)}.
\end{aligned}
$$

Thus, we can write F_{V2} as

$$
\begin{aligned}
F_{V2} &= \int q(\mathbf{Z}) q(\mathfrak{F}, \widetilde{\mathcal{F}}, \mathbf{H} | \mathbf{Z}) (\log \boldsymbol{\Delta})\, d\mathfrak{F} d\widetilde{\mathcal{F}} d\mathbf{H} d\mathbf{Z} \\
&= \int q(\mathbf{Z}) \left[\int \prod_{k=1}^{K} \phi(\boldsymbol{\eta}_k) \left\{ \log G(\mathbf{Z}, \mathbf{H}, \mathcal{Y}) + \sum_{k=1}^{K} \log \left[\frac{\Pr(\boldsymbol{\eta}_k)}{\phi(\boldsymbol{\eta}_k)} \right] \right\} d\mathbf{H} \right] d\mathbf{Z},
\end{aligned}
$$

where

$$
\log G(\mathbf{Z}, \mathbf{H}, \mathcal{Y}) = \int \Pr(\widetilde{\mathcal{F}} | \mathfrak{F}, \mathbf{Z}) \Pr(\mathfrak{F} | \mathbf{H}) \log \left[\prod_{j=1}^{M} \prod_{k=1}^{K} \left[\frac{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j)} \right]^{z_{jk}} \right] d\mathfrak{F} d\widetilde{\mathcal{F}}.
$$

We show below that $\log G(\mathbf{Z}, \mathbf{H}, \mathcal{Y})$ can be decomposed as

$$\log G(\mathbf{Z}, \mathbf{H}, \mathcal{Y}) = \sum_{j=1}^{M} \sum_{k=1}^{K} z_{jk} \log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j),$$

where

$$\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j) = \log \left[\mathcal{N}(\boldsymbol{y}^j | \boldsymbol{\alpha}_j^k, \widehat{\mathbf{K}}_{jj}) \right] - \frac{1}{2} \mathbf{Tr} \left[(\mathbf{K}_{jj}^k - \mathbf{Q}_{jj}^k) \widehat{\mathbf{K}}_{jj}^{-1} \right], \tag{4.21}$$

where $\boldsymbol{\alpha}_j^k = \mathbf{K}_{jk} \mathbf{K}_{kk}^{-1} \boldsymbol{\eta}_k$ and $\mathbf{Q}_{jj}^k = \mathbf{K}_{jk} \mathbf{K}_{kk}^{-1} \mathbf{K}_{kj}$. Consequently, the variational lower bound is

$$\begin{aligned} F_V = & \int q(\mathbf{Z}) q(\pi) \log \left[\frac{\Pr(\boldsymbol{\pi}) \Pr(\mathbf{Z}|\boldsymbol{\pi})}{q(\mathbf{Z}) q(\boldsymbol{\pi})} \right] d\boldsymbol{\pi} d\mathbf{Z} \\ & + \int q(\mathbf{Z}) \left[\int \prod_{k=1}^{K} \phi(\boldsymbol{\eta}_k) \left\{ \log G(\mathbf{Z}, \mathbf{H}, \mathcal{Y}) + \sum_{k=1}^{K} \log \left[\frac{\Pr(\boldsymbol{\eta}_k)}{\phi(\boldsymbol{\eta}_k)} \right] \right\} d\mathbf{H} \right] d\mathbf{Z} \end{aligned}$$

To optimize the parameters we use the variational EM algorithm.

- In the **Variational E-Step**, we estimate $q^*(\mathbf{Z}), q^*(\boldsymbol{\pi})$ and $\{\phi^*(\boldsymbol{\eta}_k)\}$.

  To get the variational distribution $q^*(\mathbf{Z})$, we take derivative of F_V w.r.t. $q(\mathbf{Z})$ and set it to 0. This yields

$$\begin{aligned} \log q^*(\mathbf{Z}) &= \int q(\boldsymbol{\pi}) \log(\Pr(\mathbf{Z}|\boldsymbol{\pi})) d\boldsymbol{\pi} + \int \prod_{k=1}^{K} \phi(\boldsymbol{\eta}_k) \log G(\mathbf{Z}, \mathbf{H}, \mathcal{Y}) d\mathbf{H} \\ &= \sum_{j=1}^{M} \sum_{k=1}^{K} z_{jk} \left[\mathbb{E}_{q(\boldsymbol{\pi})}[\log \pi_k] + \mathbb{E}_{\phi(\boldsymbol{\eta}_k)}[\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)] \right] + \mathbf{const} \end{aligned}$$

  from which (see similar derivation in (Bishop, 2006, chap. 9)) we obtain

$$\begin{aligned} q^*(\mathbf{Z}) &= \prod_{j=1}^{M} \prod_{k=1}^{K} r_{jk}^{z_{jk}}, \qquad r_{jk} = \frac{\rho_{jk}}{\sum_{k=1}^{K} \rho_{jk}} \\ \log \rho_{jk} &= \mathbb{E}_{q(\boldsymbol{\pi})}[\log \pi_k] + \mathbb{E}_{\phi(\boldsymbol{\eta}_k)}[\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)], \end{aligned} \tag{4.22}$$

  where $\mathbb{E}_{q(\boldsymbol{\pi})}[\log \pi_k] = \psi(\alpha_k) - \psi(\sum_k \alpha_k)$ where ψ is the digamma function, α_k

is defined below in (4.24), and $\mathbb{E}_{\phi(\boldsymbol{\eta}_k)}[\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)]$ is given below in (4.37).

For the variational distribution of $q^*(\boldsymbol{\pi})$ the derivative yields

$$\begin{aligned}\log q^*(\boldsymbol{\pi}) &= \log \Pr(\boldsymbol{\pi}) + \int q(\mathbf{Z}) \log(\Pr(\mathbf{Z}|\boldsymbol{\pi}))d\boldsymbol{\pi} + \textbf{const} \\ &= (\alpha_0 - 1)\sum_{k=1}^{K} \log(\pi_k) + \sum_{j=1}^{M}\sum_{k=1}^{K} \mathbb{E}[z_{jk}] \log \pi_k + \textbf{const},\end{aligned}$$

where we have used the form of the Dirichlet distribution. Taking the exponential of both sides, we have

$$q^*(\boldsymbol{\pi}) = \textbf{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \tag{4.23}$$

where

$$\alpha_k = \alpha_0 + N_k, \quad N_k = \sum_{j=1}^{M} r_{jk}. \tag{4.24}$$

The final step is to get the variational distribution of $\phi^*(\boldsymbol{\eta}_k), k = 1, \cdots, K$. Notice that only F_{V2} is a function of $\phi(\boldsymbol{\eta}_k)$. We can rewrite this portion as

$$\begin{aligned}&\int \prod_{k=1}^{K} \phi(\boldsymbol{\eta}_k) \left(\left\{ \int q(\mathbf{Z}) \sum_{j=1}^{M}\sum_{k=1}^{K} z_{jk} \left[\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)\right] d\mathbf{Z} \right\} + \sum_{k=1}^{K} \log\left[\frac{\Pr(\boldsymbol{\eta}_k)}{\phi(\boldsymbol{\eta}_k)}\right] \right) d\mathbf{H} \\ &= \int \prod_{k=1}^{K} \phi(\boldsymbol{\eta}_k) \left(\sum_{j=1}^{M}\sum_{k=1}^{K} \mathbb{E}_{q(\mathbf{Z})}[z_{jk}] \left[\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)\right] + \sum_{k=1}^{K} \log\left[\frac{\Pr(\boldsymbol{\eta}_k)}{\phi(\boldsymbol{\eta}_k)}\right] \right) d\mathbf{H} \\ &= \sum_{k=1}^{K} \int \phi(\boldsymbol{\eta}_k) \left\{ \left[\sum_{j=1}^{M} \mathbb{E}_{q(\mathbf{Z})}[z_{jk}] \log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j) \right] + \log\left[\frac{\Pr(\boldsymbol{\eta}_k)}{\phi(\boldsymbol{\eta}_k)}\right] \right\} d\boldsymbol{\eta}_k.\end{aligned} \tag{4.25}$$

Thus, our task reduces to find $\phi^*(\boldsymbol{\eta}_k)$ separately. Taking the derivative of (4.25) w.r.t. $\phi(\boldsymbol{\eta}_k)$ and setting it to be zero, we have

$$\log \phi^*(\boldsymbol{\eta}_k) = \sum_{j=1}^{M} \mathbb{E}_{q(\mathbf{Z})}[z_{jk}] \log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j) + \log \Pr(\boldsymbol{\eta}_k) + \textbf{const.}$$

Using (4.21) and the fact that second term in (4.21) is not a function of $\boldsymbol{\eta}_k$, we

obtain

$$\phi^*(\boldsymbol{\eta}_k) \propto \prod_{j=1}^{M} \left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{K}}_{jj}^k)\right]^{\mathbb{E}_{q(\mathbf{Z})}[z_{jk}]} \Pr(\boldsymbol{\eta}_k). \tag{4.26}$$

Thus, we have $\phi^*(\boldsymbol{\eta}_k)$ proportional to

$$\exp\left\{-\frac{1}{2}(\boldsymbol{\eta}^k)^{\mathrm{T}}\left(\mathbf{K}_{kk}^{-1}\boldsymbol{\Phi}\mathbf{K}_{kk}^{-1}\right)\boldsymbol{\eta}^k + (\boldsymbol{\eta}_k)^{\mathrm{T}}\left(\mathbf{K}_{kk}^{-1}\sum_{j=1}^{M}\mathbb{E}_{q(\mathbf{Z})}[z_{jk}]\mathbf{K}_{kj}[\widehat{\mathbf{K}}_{jj}]^{-1}\boldsymbol{y}_j\right)\right\},$$

where

$$\boldsymbol{\Phi} = \mathbf{K}_{kk} + \sum_{j=1}^{M} r_{jk}\mathbf{K}_{kj}[\widehat{\mathbf{K}}_{jj}]^{-1}\mathbf{K}_{jk}.$$

Completing the square yields the Gaussian distribution $\phi^*(\boldsymbol{\eta}_k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where

$$\boldsymbol{\mu}_k = \mathbf{K}_{kk}\boldsymbol{\Phi}^{-1}\sum_{j=1}^{M} r_{jk}\mathbf{K}_{kj}[\widehat{\mathbf{K}}_{jj}]^{-1}\boldsymbol{y}_j, \quad \boldsymbol{\Sigma}_k = \mathbf{K}_{kk}\boldsymbol{\Phi}^{-1}\mathbf{K}_{kk}. \tag{4.27}$$

- In the **Variational M-Step**, based on the previous estimated variational distribution, we wish to find hyperparameters that maximize the variational lower bound F_V. The terms that depend on the hyperparameters and the inducing variables $\{\mathcal{X}_m^k\}$ are given in (4.25). Therefore, using (4.21) again, we have

$$\begin{aligned}
F_V(\mathcal{X}_k, \boldsymbol{\theta}) &= \sum_{k=1}^{K}\int \phi^*(\boldsymbol{\eta}_k)\left\{\left[\sum_{j=1}^{M} r_{jk}\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j)\right] + \log\left[\frac{\Pr(\boldsymbol{\eta}_k)}{\phi^*(\boldsymbol{\eta}_k)}\right]\right\} d\boldsymbol{\eta}_k \\
&= \sum_{k=1}^{K}\int \phi^*(\boldsymbol{\eta}_k)\left\{\log\left[\sum_{j=1}^{M} r_{jk}\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{K}}_{jj})\right] + \log\left[\frac{\Pr(\boldsymbol{\eta}_k)}{\phi^*(\boldsymbol{\eta}_k)}\right]\right\} d\boldsymbol{\eta}_k \\
&\qquad - \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{m} r_{jk}\mathbf{Tr}\left[(\mathbf{K}_{jj}^k - \mathbf{Q}_{jj}^k)\widehat{\mathbf{K}}_{jj}^{-1}\right] \\
&= \sum_{k=1}^{K}\int \phi^*(\boldsymbol{\eta}_k)\left\{\log\left[\frac{\prod_{j=1}^{M}\left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{K}}_{jj})\right]^{r_{jk}}\Pr(\boldsymbol{\eta}_k)}{\phi^*(\boldsymbol{\eta}_k)}\right]\right\} d\boldsymbol{\eta}_k \\
&\qquad - \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{m} r_{jk}\mathbf{Tr}\left[(\mathbf{K}_{jj}^k - \mathbf{Q}_{jj}^k)\widehat{\mathbf{K}}_{jj}^{-1}\right]
\end{aligned}$$

From (4.26), we know that the term inside the log is constant, and therefore,

extracting the log from the integral and cancelling the $\phi*$ terms we see that the k'th element of first term is equal to the logarithm of

$$\int \prod_{j=1}^{M} \left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{K}}_{jj})\right]^{r_{jk}} \Pr(\boldsymbol{\eta}_k) d\boldsymbol{\eta}_k. \tag{4.28}$$

We next show how this multivariate integral can be evaluated. First consider

$$\begin{aligned}
&\left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{K}}_{jj})\right]^{r_{jk}} \\
&= \left((2\pi)^{-\frac{N_j}{2}} |\widehat{\mathbf{K}}_{jj}|^{-\frac{1}{2}}\right)^{r_{jk}} \exp\left\{-\frac{r_{jk}}{2}(\boldsymbol{y}^j - \boldsymbol{\alpha}_j^k)^{\mathrm{T}}[\widehat{\mathbf{K}}_{jj}]^{-1}(\boldsymbol{y}^j - \boldsymbol{\alpha}_j^k)\right\} \\
&= \left((2\pi)^{-\frac{N_j}{2}} |\widehat{\mathbf{K}}_{jj}|^{-\frac{1}{2}}\right)^{r_{jk}} \exp\left\{-\frac{1}{2}(\boldsymbol{y}^j - \boldsymbol{\alpha}_j^k)^{\mathrm{T}}\left[r_{jk}^{-1}\widehat{\mathbf{K}}_{jj}\right]^{-1}(\boldsymbol{y}^j - \boldsymbol{\alpha}_j^k)\right\} \\
&= \frac{\left((2\pi)^{-\frac{N_j}{2}} |\widehat{\mathbf{K}}_{jj}|^{-\frac{1}{2}}\right)^{r_{jk}}}{(2\pi)^{-\frac{N_j}{2}} |r_{jk}^{-1}\widehat{\mathbf{K}}_{jj}|^{-\frac{1}{2}}} \cdot (2\pi)^{-\frac{N_j}{2}} |r_{jk}^{-1}\widehat{\mathbf{K}}_{jj}|^{-\frac{1}{2}} \\
&\quad \times \exp\left\{-\frac{1}{2}(\boldsymbol{y}^j - \boldsymbol{\alpha}_j^k)^{\mathrm{T}}\left[r_{jk}^{-1}\widehat{\mathbf{K}}_{jj}\right]^{-1}(\boldsymbol{y}^j - \boldsymbol{\alpha}_j^k)\right\} = A_{jk}\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, r_{jk}^{-1}\widehat{\mathbf{K}}_{jj}^k),
\end{aligned}$$

where $A_{jk} = (r_{jk})^{\frac{N_j}{2}}(2\pi)^{\frac{N_j(1-r_{jk})}{2}}|\widehat{\mathbf{K}}_{jj}|^{\frac{1-r_{jk}}{2}}$. Thus, we have

$$\prod_{j=1}^{M}\left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{K}}_{jj})\right]^{r_{jk}} = \left[\prod_{j=1}^{M} A_{jk}\right]\prod_{j=1}^{M}\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, r_{jk}^{-1}\widehat{\mathbf{K}}_{jj}).$$

The first part is not a function of $\boldsymbol{\eta}_k$ and therefore, for the integration we are only interested in the second part. Since $\mathcal{Y}$ is the concatenation of all $\boldsymbol{y}^j$'s, we can write

$$\prod_{j=1}^{M}\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, r_{jk}^{-1}\widehat{\mathbf{K}}_{jj}) = \mathcal{N}(\mathcal{Y}|\boldsymbol{\Lambda}_k\mathbf{K}_{kk}^{-1}\boldsymbol{\eta}_k, \widehat{\mathbf{K}}^k), \tag{4.29}$$

where

$$\boldsymbol{\Lambda}_k = \begin{pmatrix} \mathbf{K}_{1k} \\ \mathbf{K}_{2k} \\ \vdots \\ \mathbf{K}_{Mk} \end{pmatrix} \qquad \text{and} \qquad \widehat{\mathbf{K}}^k = \bigoplus_{j=1}^{M} r_{jk}^{-1} \widehat{\mathbf{K}}_{jj}^k = \begin{pmatrix} r_{1k}^{-1}\widehat{\mathbf{K}}_{11} & & & \\ & r_{2k}^{-1}\widehat{\mathbf{K}}_{22} & & \\ & & \ddots & \\ & & & r_{Mk}^{-1}\widehat{\mathbf{K}}_{MM} \end{pmatrix}.$$

Therefore, the integral can the written as the following marginal distribution of $\Pr(\mathcal{Y}|k)$,

$$\int \prod_{j=1}^{M} \mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, r_{jk}^{-1}\widehat{\mathbf{K}}_{jj}) \Pr(\boldsymbol{\eta}_k) d\boldsymbol{\eta}_k = \int \mathcal{N}(\mathcal{Y}|\boldsymbol{\Lambda}_k \mathbf{K}_{kk}^{-1}\boldsymbol{\eta}_k, \widehat{\mathbf{K}}^k) \Pr(\boldsymbol{\eta}_k) d\boldsymbol{\eta}_k = \Pr(\mathcal{Y}|k). \tag{4.30}$$

Using the fact that $\Pr(\boldsymbol{\eta}_k) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{kk})$ and observing that (4.29) is a conditional Gaussian, we have

$$\Pr(\mathcal{Y}|k) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}_k \mathbf{K}_{kk}^{-1} \boldsymbol{\Lambda}_k^{\mathrm{T}} + \widehat{\mathbf{K}}^k).$$

Using this form and the portion of A_{jk} that depends on the parameters we obtain the variational lower bound $F_V(\mathcal{X}, \boldsymbol{\theta})$,

$$\begin{aligned} &\sum_{k=1}^{K} \log \Pr(\mathcal{Y}|k) + \sum_{k=1}^{K}\sum_{j=1}^{M} \frac{1 - r_{jk}}{2} \log|\widehat{\mathbf{K}}_{jj}| - \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{M} r_{jk} \mathbf{Tr}\left[(\mathbf{K}_{jj}^k - \mathbf{Q}_{jj}^k)\widehat{\mathbf{K}}_{jj}^{-1}\right] \\ &= \sum_{k=1}^{K} \log \Pr(\mathcal{Y}|k) + \frac{K-1}{2}\sum_{j=1}^{M} \log|\widehat{\mathbf{K}}_{jj}| - \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{M} r_{jk} \mathbf{Tr}\left[(\mathbf{K}_{jj}^k - \mathbf{Q}_{jj}^k)\widehat{\mathbf{K}}_{jj}^{-1}\right] \end{aligned} \tag{4.31}$$

This extends the bound for the single center $K = 1$ case given in (4.9). Furthermore, following the same reasoning as the previous derivation, the direct inference for the full model can be obtained where η_k is substituted with $\mathfrak{f}_k$ and the variational lower bound becomes

$$F_V(\mathcal{X}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \log \mathcal{N}(\mathcal{Y}|\mathbf{0}, \mathbf{K}_{kk} + \widehat{\mathbf{K}}^k) + \frac{K-1}{2}\sum_{j=1}^{M} \log|\widehat{\mathbf{K}}_{jj}|. \tag{4.32}$$

We have explicitly written the parameters that can be chosen to further optimize the lower bound (4.31), namely the support inputs $\{\mathcal{X}_m^k\}$, and the hyper-parameters $\boldsymbol{\theta}$ which are composed of $\{\boldsymbol{\theta}_k\}$ and $\{\tilde{\boldsymbol{\theta}}\}$ in K_k and $\widetilde{K}$ respectively.

By calculating derivatives of (4.31) we can optimize the lower bound using a gradient based method. It is easy to see that the complexity for calculating the derivative of the second and third terms of (4.31) is $\mathcal{O}(N)$. Thus, the key computational issue of deriving a gradient descent algorithm involves computing the derivative of $\log \Pr(\mathcal{Y}|k)$. We first show how to calculate the inverse of the $N \times N$ matrix $\mathbf{Y} = \boldsymbol{\Lambda}_k \mathbf{K}_{kk}^{-1} \boldsymbol{\Lambda}_k^{\mathrm{T}} + \widehat{\mathbf{K}}^k$. Using the matrix inversion lemma (the Woodbury identity (Bishop, 2006)), we have

$$(\boldsymbol{\Lambda}_k \mathbf{K}_{kk}^{-1} \boldsymbol{\Lambda}_k^{\mathrm{T}} + \widehat{\mathbf{K}}^k)^{-1} = [\widehat{\mathbf{K}}^k]^{-1} - [\widehat{\mathbf{K}}^k]^{-1} \boldsymbol{\Lambda}_k \left(\mathbf{K}_{kk} + \boldsymbol{\Lambda}_k^{\mathrm{T}} [\widehat{\mathbf{K}}^k]^{-1} \boldsymbol{\Lambda}_k \right)^{-1} \boldsymbol{\Lambda}_k^{\mathrm{T}} [\widehat{\mathbf{K}}^k]^{-1}.$$

Since $\widehat{\mathbf{K}}^k$ is a block-diagonal matrix, its inverse can be calculated in $\sum_j \mathcal{O}(N_j^3)$. Now, $\mathbf{K}_{kk} + \boldsymbol{\Lambda}_k^{\mathrm{T}} [\widehat{\mathbf{K}}^k]^{-1} \boldsymbol{\Lambda}_k$ is an $m_k \times m_k$ matrix where m_k is the number of inducing variables for the k-th mean effect. Therefore the computation of (4.31) can be done in $\mathcal{O}(m_k^3 + \sum_j N_j^3 + N m_k^2)$. Next, consider calculating the derivative of the first term. We have

$$\frac{\partial \Pr(\mathcal{Y}|k)}{\partial \theta_j} = \frac{1}{2} \mathcal{Y}^{\mathrm{T}} \mathbf{Y}^{-1} \frac{\partial \mathbf{Y}}{\partial \theta_j} \mathbf{Y}^{-1} \mathcal{Y} - \frac{1}{2} \mathbf{Tr}(\mathbf{Y}^{-1} \frac{\partial \mathbf{Y}}{\partial \theta_j}),$$

where, by the chain rule, we have

$$\frac{\partial \mathbf{Y}}{\partial \theta_j} = \frac{\partial \boldsymbol{\Lambda}_k}{\partial \theta_j} \mathbf{K}_{kk}^{-1} \boldsymbol{\Lambda}_k^{\mathrm{T}} - {}_k \mathbf{K}_{kk}^{-1} \frac{\partial \mathbf{K}_{kk}}{\partial \theta_j} \mathbf{K}_{kk}^{-1} \boldsymbol{\Lambda}_k^{\mathrm{T}} + \boldsymbol{\Lambda}_k \mathbf{K}_{kk}^{-1} \frac{\partial \boldsymbol{\Lambda}_k^{\mathrm{T}}}{\partial \theta_j} + \frac{\partial \widehat{\mathbf{K}}^k}{\partial \theta_j}.$$

Therefore, pre-calculating $\mathcal{Y}^{\mathrm{T}} \mathbf{Y}^{-1}$ and sequencing the other matrix operations from left to right the gradient calculation for each hyperparameter can be calculated in $\mathcal{O}(N m_k^2)$. In our implementation, we use stochastic coordinate descent, where at each iteration, one coordinate (parameter) is chosen at random and we perform gradient descent on that coordinate.

Evaluating $\log G(\mathbf{Z}, \mathbf{H}, \mathcal{Y})$

In this section, we develop the expression for $\log G(\mathbf{Z}, \mathbf{H}, \mathcal{Y})$.

$$
\begin{aligned}
&\log G(\mathbf{Z}, \mathbf{H}, \mathcal{Y}) \\
&= \int \prod_{l,p} \left[\Pr(\tilde{\mathbf{f}}^l | \mathfrak{f}_p, \boldsymbol{y}^l) \right]^{z_{lp}} \prod_{v=1}^{K} \Pr(\mathfrak{f}_v | \boldsymbol{\eta}_v) \times \sum_{j=1}^{M} \sum_{k=1}^{K} z_{jk} \log \left[\frac{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j)} \right] d\mathfrak{F} d\widetilde{\mathcal{F}} \\
&= \sum_{j,k} z_{jk} \left[\int \left(\prod_{v=1}^{K} \Pr(\mathfrak{f}_v | \boldsymbol{\eta}_v) \right) \times \Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j) \times \log \left[\frac{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j)} \right] d\mathfrak{F} d\tilde{\mathbf{f}}^j \right] \\
&= \sum_{j,k} z_{jk} \Bigg[\int \Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j) \\
&\qquad \times \left[\int \Pr(\mathfrak{f}_k | \boldsymbol{\eta}_k) \prod_{v=1, v \neq k}^{K} \Pr(\mathfrak{f}_v | \boldsymbol{\eta}_v) d\mathfrak{F}_{-k} \right] \times \log \left[\frac{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j)} \right] d\mathfrak{f}_k \tilde{\mathbf{f}}^j \Bigg] \\
&= \sum_{j,k} z_{jk} \left[\int \Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j) \Pr(\mathfrak{f}_k | \boldsymbol{\eta}_k) \times \log \left[\frac{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j)} \right] d\mathfrak{f}_k d\tilde{\mathbf{f}}^j \right] \\
&= \sum_{j,k} z_{jk} \log G(\boldsymbol{\eta}_k, \boldsymbol{y}_j)
\end{aligned}
\tag{4.33}
$$

where the second line holds because in the sum indexed by j and k all the product measures

$$
\prod_{l=1, l \neq j}^{M} \prod_{p=1}^{K} \left[\Pr(\tilde{\mathbf{f}}^l | \mathfrak{f}_p, \boldsymbol{y}^l) \right]^{z_{lp}},
$$

are integrated to 1, leaving only the $\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j)$. Our next step is to evaluate $\log G(\boldsymbol{\eta}_k, \boldsymbol{y}_j)$, we have

$$
\begin{aligned}
&\int \Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j) \Pr(\mathfrak{f}_k | \boldsymbol{\eta}_k) \times \log \left[\frac{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j)}{\Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j)} \right] d\mathfrak{f}_k d\tilde{\mathbf{f}}^j \\
&= \int \Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k, \boldsymbol{y}^j) \Pr(\mathfrak{f}_k | \boldsymbol{\eta}_k) \times \log \left[\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j) \cdot \frac{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k)}{\Pr(\boldsymbol{y}^j | \mathfrak{f}_k, \tilde{\mathbf{f}}^j) \Pr(\tilde{\mathbf{f}}^j | \mathfrak{f}_k)} \right] d\mathfrak{f}_k d\tilde{\mathbf{f}}^j \\
&= \int \Pr(\mathfrak{f}_k | \boldsymbol{\eta}_k) \log \left[\Pr(\boldsymbol{y}^j | \mathfrak{f}_k) \right] d\mathfrak{f}_k \qquad (4.34) \\
&= \int \Pr(\mathbf{f}^j | \boldsymbol{\eta}_k) \log \left[\Pr(\boldsymbol{y}^j | \mathbf{f}^j) \right] d\mathbf{f}^j \qquad (4.35)
\end{aligned}
$$

where the one to last line holds because of the independence between $\tilde{\mathbf{f}}^j$ and f_k. We next show how this expectation can be evaluated. This is the same derivation as in the single center case (Section 4.2), but we repeat it here using the appropriate notation for completeness.

Recall that $\Pr(\mathbf{f}^j|\boldsymbol{\eta}_k) = \mathcal{N}(\mathbf{f}^j|\mathbf{K}_{jk}\mathbf{K}_{kk}^{-1}\boldsymbol{\eta}_k, \mathbf{K}_{jj}^k - \mathbf{K}_{jk}\mathbf{K}_{kk}^{-1}\mathbf{K}_{kj})$. Denote $\widehat{\mathbf{K}}_{jj}^{-1} = \mathbf{L}^{\mathrm{T}}\mathbf{L}$ where $\mathbf{L}$ can be chosen as its Cholesky factor, we have

$$\log\left[\Pr(\boldsymbol{y}^j|\mathbf{f}^j)\right] = -\frac{1}{2}(\mathbf{L}\boldsymbol{y}^j - \mathbf{L}\mathbf{f}^j)^{\mathrm{T}}(\mathbf{L}\boldsymbol{y}^j - \mathbf{L}\mathbf{f}^j) + \log\left[(2\pi)^{-\frac{N_j}{2}}\right] + \log\left[|\widehat{\mathbf{K}}_{jj}|^{-\frac{1}{2}}\right].$$

Notice that $\Pr(\mathbf{L}\mathbf{f}^j|\boldsymbol{\eta}_k) = \mathcal{N}(\mathbf{L}\mathbf{K}_{jk}\mathbf{K}_{kk}^{-1}\boldsymbol{\eta}_k, \mathbf{L}(\mathbf{K}_{jj}^k - \mathbf{Q}_{jj}^k)\mathbf{L}^{\mathrm{T}})$ where $\mathbf{Q}_{jj}^k = \mathbf{K}_{jk}\mathbf{K}_{kk}^{-1}\mathbf{K}_{kj}$. Thus, we have

$$\begin{aligned}
\log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j) &= \mathbb{E}_{[\mathbf{f}^j|\boldsymbol{\eta}_k]} \log\left[\Pr(\boldsymbol{y}^j|\mathbf{f}^j)\right] \\
&= -\frac{1}{2}\left\|\mathbf{L}\boldsymbol{y}^j - \mathbf{L}\mathbf{K}_{jk}\mathbf{K}_{kk}^{-1}\boldsymbol{\eta}_k\right\|^2 \\
&\quad - \frac{1}{2}\mathbf{Tr}(\mathbf{L}(\mathbf{K}_{jj}^k - \mathbf{Q}_{jj}^k)\mathbf{L}^{\mathrm{T}}) + \log\left[(2\pi)^{-\frac{N_j}{2}}\right] + \log\left[|\widehat{\mathbf{K}}_{jj}|^{-\frac{1}{2}}\right] \\
&= \log\left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j, \widehat{\mathbf{K}}_{jj})\right] - \frac{1}{2}\mathbf{Tr}\left[(\mathbf{K}_{jj}^k - \mathbf{Q}_{jj}^k)\widehat{\mathbf{K}}_{jj}^{-1}\right]
\end{aligned}$$

where $\boldsymbol{\alpha}_j^k = \mathbf{K}_{jk}\mathbf{K}_{kk}^{-1}\boldsymbol{\eta}_k$. Finally, we have

$$\log G(\mathbf{H}, \mathcal{Y}) = \sum_{j=1}^{m}\sum_{k=1}^{K} z_{jk}\left[\log\left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\alpha}_j^k, \widehat{\mathbf{K}}_{jj})\right] - \frac{1}{2}\mathbf{Tr}\left[(\mathbf{K}_{jj}^k - \mathbf{Q}_{jj}^k)[\widehat{\mathbf{K}}_{jj}]^{-1}\right]\right]. \quad (4.36)$$

Furthermore, marginalization out $\boldsymbol{\eta}_k$, we have

$$\begin{aligned}
&\mathbb{E}_{\phi^*(\boldsymbol{\eta}_k)} \log G(\boldsymbol{\eta}_k, \boldsymbol{y}^j) \\
&\quad = \log\left[\mathcal{N}(\boldsymbol{y}^j|\boldsymbol{\mu}_j^k, \widehat{\mathbf{K}}_{jj})\right] - \frac{1}{2}\mathbf{Tr}\left[\mathbf{K}_{jk}\mathbf{K}_{kk}^{-1}(\boldsymbol{\Sigma}_k - \mathbf{K}_{kk})\mathbf{K}_{kk}^{-1}\mathbf{K}_{jk}\widehat{\mathbf{K}}_{jj}^{-1}\right].
\end{aligned} \quad (4.37)$$

4.3.3 Algorithm Summary

The various steps in our algorithm and their time complexity are summarized in Algorithm 2.

Algorithm 2 VARIATIONAL EM ALGORITHM FOR SPARSE GMT

1: Initialize $\mathcal{X}$ and the hyper parameters $\boldsymbol{\theta}$;
2: **repeat**
3: Calculate $\{r_{jk}\}$ using (4.22) with time complexity $\mathcal{O}(KNm_k^2)$
4: Estimate $q^*(\mathbf{Z})$ using (4.22) with time complexity $\mathcal{O}(MK)$;
5: Estimate $q^*(\boldsymbol{\pi})$ using (4.23) with time complexity $\mathcal{O}(MK)$;
6: Estimate $\{\phi^*(\boldsymbol{\eta}_k)\}$ using (4.27) with time complexity $\mathcal{O}(KNm_k^2)$;
7: Use the stochastic coordinate descent to optimize the hyperparameters and pseudo inputs. For this, calculate the derivatives of the variational lower bound (4.31) that can be done in $\mathcal{O}(KNm_k^2)$.
8: **until** converges or reach the iteration limit

4.3.4 Prediction Using the Sparse Model

The proposed sparse model can be used for two types of problems. Prediction for existing tasks and prediction for a newly added task. We start with deriving the predictive distribution for existing tasks. Given any task j, our goal is to calculate the predictive distribution $\Pr(f^j(\boldsymbol{x}^*)|\mathcal{Y})$ at new input point $\boldsymbol{x}^*$, which can be written as

$$\sum_{k=1}^{K} \Pr(f^j(\boldsymbol{x}^*)|z_{jk}=1,\mathcal{Y}) \Pr(z_{jk}=1|\mathcal{Y}) = \sum_{k=1}^{K} r_{jk} \Pr(f^j(\boldsymbol{x}^*)|z_{jk}=1,\mathcal{Y}). \tag{4.38}$$

That is, because z_{jk} form a partition we can focus on calculating $\Pr(f^j(\boldsymbol{x}^*)|z_{jk}=1,\mathcal{Y})$ and then combine the results using the partial labels. Calculating (4.38) is exactly the same as the predictive distribution in the non-grouped case, the derivation in Section 4.2 gives the details. The complexity of these computations is $\mathcal{O}(K(N_j^3+m^3))$ which is a significant improvement over $\mathcal{O}(KN^3)$ where $N=\sum_j N_j$. Instead of calculating the full Bayesian prediction, one can use *Maximum A Posteriori* (MAP) by assigning the j-th task to the center c such that $c = \operatorname{argmax} \Pr(z_{jk}=1|\mathcal{Y})$. Preliminary experiments (not shown here) show that the full Bayesian approach gives better performance. Our experiment below uses the results of Bayesian prediction.

Our model is also useful for making prediction for newly added tasks. Suppose we are given $\{\boldsymbol{x}^{M+1}, \boldsymbol{y}^{M+1}\}$ and we are interested in predicting $f^{M+1}(\boldsymbol{x}^*)$. We use the variational procedure to estimate its partial labels w.r.t. different centers $\Pr(z_{M+1,k} = 1|\mathcal{Y})$ and then (4.38) can be applied for making the prediction. In the variational procedure we update the parameters for Z_{M+1} but keep all other parameters fixed. Since each task has small number of samples, we expect this step to be computationally cheap.

4.4 Experiments

For performance criteria we use the standardized mean square error (SMSE) and the mean standardized log loss (MSLL) that are defined in (Rasmussen and Williams, 2006). We compare the following methods. The first four methods use the same variational inference as describe in Section 4.2. They differ in the form of the variational lower bound they choose to optimize.

1. **Direct Inference**: use full samples as the support variables and optimize the marginal likelihood. When $k = 1$, the marginal likelihood is described in Section 4.2 and the predictive distribution is (4.4). For general k, F_v is given in (4.32).

2. **Variational Sparse GMT (MT-VAR)**: the proposed approach.

3. **MTL Subset of Datapoints (MT-SD)**: a subset $\mathcal{X}_m^k$ of size m_k is chosen uniformly from the input points from all tasks $\mathcal{X}$ for each center. The hyperparameters are selected using $\mathcal{X}_m^k$ (the inducing variables are fixed in advance) and their corresponding observations by maximizing the variational lower bound. We call this MT-SD as a multi-task version of SD (see Rasmussen and Williams, 2006, Chap. 8.3.2), because in the single center case, this method uses (4.3) and (4.4) using the subset $\mathcal{X}_m, \mathcal{Y}_m$ and $\boldsymbol{x}^j, \boldsymbol{y}^j$ as the full sample (thus discarding other samples).

4. **MTL Projected Process Approximation (MT-PP)**: the variational lower bound of MT-PP is given by the first two terms of (4.31) ignoring the trace term, and therefore the optimization chooses different pseudo inputs and hyper-parameters. We call this method MT-PP because in the single center case, it corresponds to a multi-task version of PP (see Rasmussen and Williams, 2006, chap. 8.3.3).

5. **Convolved Multiple Output GP (MGP-FITC, MGP-PITC)**: the approaches proposed in (Álvarez and Lawrence, 2011). For all experiments, we use code from (Álvarez and Lawrence, 2011) with the following setting. The kernel type is set to be gg. The hyper-parameters, parameters and the position of inducing variables are obtained via optimizing the marginal likelihood using a scaled conjugate gradient algorithm. The support variables are initialized as equally spaced points over the range of the inputs. We set the parameter $R_q = 1$, which means that the latent functions share the same covariance function. Whenever possible, we set Q which, roughly speaking, corresponds to the number of centers in our approach, to agree with the number of centers. The maximum number of iterations allowed in the optimization procedure is set to be 200. The number of support variables is controlled in the experiments as in our methods.

Three datasets are used to demonstrate the empirical performance of the proposed approach. The first synthetic dataset contains data sampled according to our model. The second dataset is also synthetic but it is generated from differential equations describing glucose concentration in biological experiments, a problem that has been previously used to evaluate multi-task GP (Pillonetto et al., 2010). Finally, we apply the proposed method on the astrophysics dataset described in the previous chapter.

As in the previous chapter, for all experiments, the kernels for different centers are assumed to be the same. The hyperparameter for the Dirichlet distribution is set to be $\alpha_0 = 1/K$. Unless otherwise specified, the inducing variables are initial-

ized to be equally spaced points over the range of the inputs. To initialize, tasks are randomly assigned into groups. We run the conjugate gradient algorithm (implemented via `minimize.m` in MATLAB) on a small subset of tasks (100 tasks each having 5 samples) to get the starting values of hyperparameters of the $\widetilde{\mathcal{K}}$ and $\mathcal{K}$, and then follow with the full optimization as above. Finally, we repeat the entire procedure 5 times and choose the one that achieves the best variational lower bound. The maximum number of iterations for the stochastic coordinate descent is set to be 50 and the maximum number of iterations for the variational inference is set to be 30. The entire experiment is repeated 10 times to obtain the average performance and error bars.

4.4.1 Synthetic data

In the first experiment, we demonstrate the performance of our algorithm on a regression task with artificial data. More precisely, we generated 1000 single-center tasks where each $f^j(x) = \bar{f}(x) + \tilde{f}^j(x)$ is generated on the interval $x \in [-10, 10]$. Each task has 5 samples. The fixed-effect function is sampled from a GP with covariance function

$$\mathbf{Cov}[\bar{f}(t_1), \bar{f}(t_2)] = e^{-(t_1-t_2)^2/2}.$$

The individual effect $\tilde{f}^j$ is sampled via a GP with the covariance function

$$\mathbf{Cov}[\tilde{f}^j(t_1), \tilde{f}^j(t_2)] = 0.25e^{-(t_1-t_2)^2/2}.$$

The noise level σ^2 is set to be 0.1. The sample points $\boldsymbol{x}^j$ for each task are sampled uniformly in the interval $[-10, 10]$ and the 100 test samples are chosen equally spaced in the same interval. The fixed-effect curve is generated by drawing a single realization from the distribution of $\bar{\mathbf{f}}$ while the $\{\mathbf{f}^j\}$ are sampled i.i.d. from their common prior. We set the number of latent functions $Q = 1$ for MGP.

The results are shown in Figure 4-2. The left column shows qualitative results for one run using 20 support variables. We restrict the initial support variables

to be in $[-7, 7]$ on purpose to show that the proposed method is capable of finding the optimal inducing variables. It is clear that the predictive distribution of the proposed method is much closer to the results of direct inference. The right column gives quantitative results for SMSE and MSLL showing the same, as well as showing that with 40 pseudo inputs the proposed method recovers the performance of full inference. The MGP performs poorly on this dataset, indicating that it is not sufficient to capture the random effect.

We note that the comparison in Figure 4-2 (and similar comparisons later in this chapter) compare the performance of the algorithms when using the same number of pseudo inputs, and not directly at the same time complexity. In terms of asymptotic run time the direct inference is $\mathcal{O}(N^3)$ and the SD method requires $\mathcal{O}(m^3)$. All other algorithms require $\mathcal{O}(Nm^2)$ and PP is closely related to the proposed method. In practice, in our experiments SD is faster when using the same m but it is significantly worse in terms of accuracy. PP and our algorithm have very close run times. On the other hand, we see a large computational advantage over MGP.

We also see a large computational advantage over MGP in this experiment. When the number of inducing variables is 20, the training time for FITC (the time for constructing the sparse model plus the time for optimization) is 1515.19 seconds whereas the proposed approach is about 7 times faster (201.81 sec.)[1].

[1]The experiment was performed using MATLAB R2012a on an Intel Core Quo 6600 powered Windows 7 PC with 4GB memory.

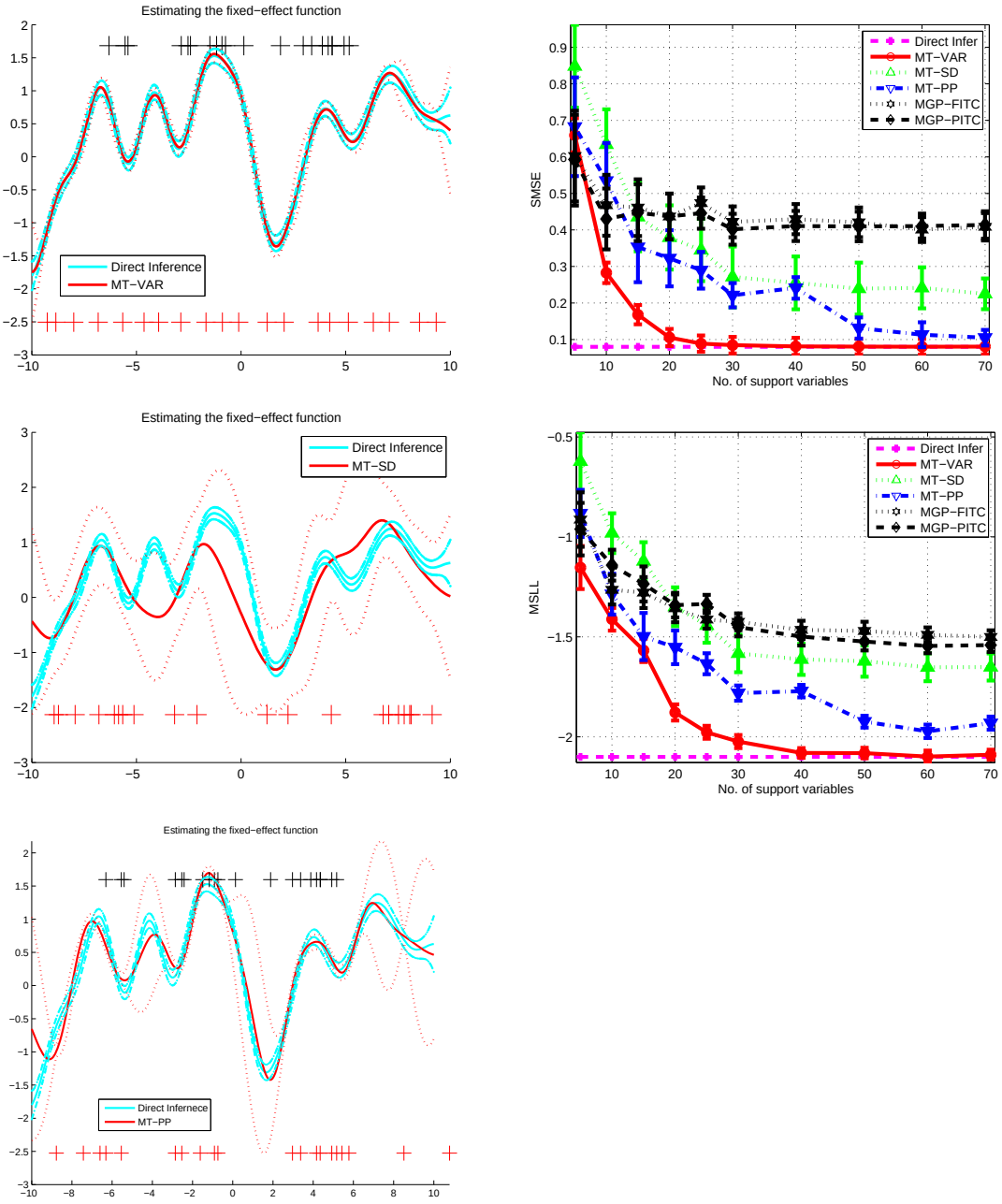

Figure 4-2: Synthetic Data: Comparison between the proposed method and other approaches. Left Column: Predictive distribution for the fixed-effect. The solid line denotes the predictive mean and the corresponding dotted line is the predictive variance. The black crosses at the top are the initial value of the inducing variables and the red ones at the bottom are their values after learning process. Right Column: The average SMSE and MSLL for all the tasks.

4.4.2 Simulated Glucose Data

We evaluate our method to reconstruct the glucose profiles in an intravenous glucose tolerance test (IVGTT) (Vicini and Cobelli, 2001; Denti et al., 2010; Pillonetto et al., 2010) where Pillonetto et al. (2010) developed an online multi-task GP solution for the case where sample points are frequently shared among tasks. This provides a more realistic test of our algorithm because data is not generated explicitly by our model. More precisely, we apply the algorithm to reconstruct the glucose profiles in an intravenous glucose tolerance test (IVGTT) where blood samples are taken at irregular intervals of time, following a single intravenous injection of glucose. We generate the data using a minimal model of glucose which is commonly used to analyze glucose and insulin IVGTT data (Vicini and Cobelli, 2001), as follows (Denti et al., 2010)

$$\begin{aligned} \dot{G}(t) &= -[S_G + X(t)]G(t) + S_G \cdot G_b + \delta(t) \cdot D/V \\ \dot{X}(t) &= -p_2 \cdot X(t) + p_2 \cdot S_I \cdot [I(t) - I_b] \\ G(0) &= G_b, \quad X(0) = 0 \end{aligned} \tag{4.39}$$

where D denotes the glucose dose, $G(t)$ is plasma glucose concentration and $I(t)$ is the plasma insulin concentration which is assumed to be known. G_b and I_b are the glucose and insulin base values. $X(t)$ is the insulin action and $\delta(t)$ is the Dirac delta function. S_G, S_I, p_2, V are four parameters of this model.

We generate 1000 synthetic subjects (tasks) following the setup in previous work: 1) the four parameters are sampled from a multivariate Gaussian with the results from the normal group in Table 1. of (Vicini and Cobelli, 2001), i.e.

$$\begin{aligned} \boldsymbol{\mu} &= [2.67, 6.42, 4.82, 1.64] \\ \boldsymbol{\Sigma} &= \mathbf{diag}(1.02, 6.90, 2.34, 0.22); \end{aligned}$$

2) $I(t)$ is obtained via spline interpolation using the real data in (Vicini and Cobelli,

2001); 3) G_b is fixed to be 84 and D is set to be 300; 4) $\delta(t)$ is simulated using a Gaussian profile with its support on the positive axis and the standard deviation (SD) randomly drawn from a uniform distribution on the interval $[0, 1]$; 5) Noise is added to the observations with $\sigma^2 = 1$. Each task has 5 measurements chosen uniformly from the interval $[1, 240]$ and 10 additional measurements are used for testing. Notice that the approach in (Pillonetto et al., 2010) cannot deal the situation efficiently since the inputs do not share samples often.

The experiments were done under both the single center and the multi center setting and the results are shown in Figure 4-3. The plots of task distribution on the top row suggest that one can get more accurate estimation by using multiple centers. For the multiple center case, the number of centers for the proposed method is arbitrarily set to be 3 ($K = 3$) and the number of latent function of MGP is set to be 2 ($Q = 2$) (We were not able of obtain reasonable results using MGP when $Q = 3$). First, we observe that the multi-center version performs better than the single center one, indicating that the group-based generalization of the traditional mixed-effect model is beneficial. Second, we can see that all the methods achieve reasonably good performance, but that the proposed method significantly outperforms the other methods.

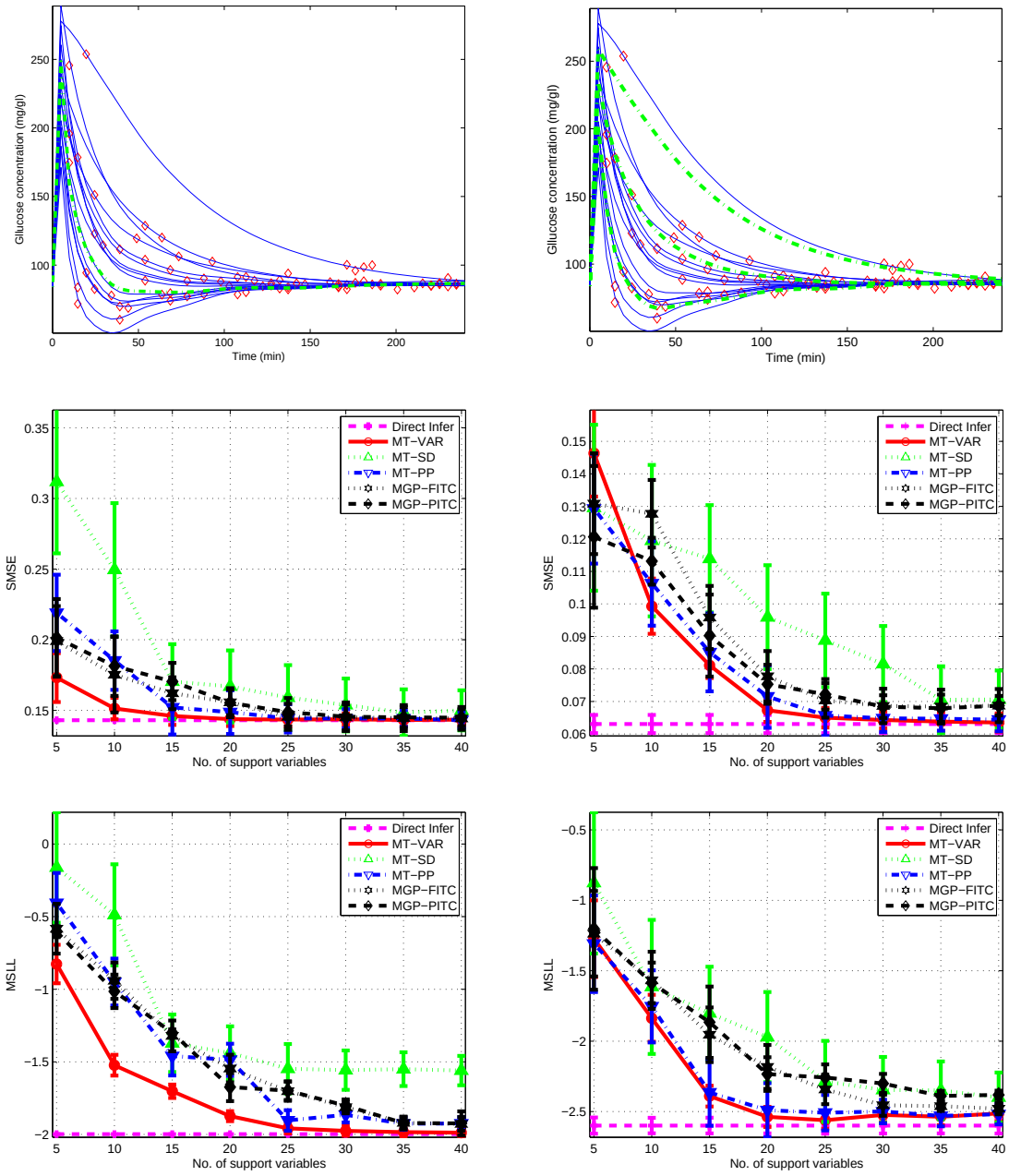

Figure 4-3: Simulated Glucose Data. Left Column: Single center $K = 1$ results; Right Column: Multiple center $K = 3$ results; Top: 15 tasks (Blue) with observations (Red Diamonds) and estimated fixed-effect curve (Green) obtained from 1000 IVGTT responses. Although the data is not generated by our model, it can be seen that different tasks have a common shape and might be modeled using a fixed effect function plus individual variations. Middle: The average SMSE for all tasks; Bottom: The average MSLL for all tasks.

4.4.3 Real Astrophysics Data

We evaluate our method again using the dataset extracted from the OGLEII survey that includes stars of 3 types (RRL, CEPH, EB) which constitute 3 datasets in our context. Here we use a random subset of 700 stars (tasks) for each type and preprocess the data normalizing each star to have mean 0 and standard deviation 1, and using universal phasing (see Section 3.4.2) to phase each time series to align the maximum of a sliding window of 5% of the original points. For each time series, we randomly sample 10 examples for training and 10 examples for testing per evaluation of SMSE and MSLL. The number of centers is set to be 3 for the proposed approach and for MGP we set $Q = 1$ (We were not able to use $Q > 1$). The results are shown in Figure 4-4. We can see that the proposed model significantly outperforms all other methods on EB. For Cepheid and RRL whose shape is simpler, we see that the error of the proposed model and of MGP are close and both outperform other methods.

Recall that in Section 3.4.2, we used a simple approach clipping sample points to a fine grid of 200 equally spaced points, due to the high dimensionality of the full sample (over 18000 points). We compare the proposed approach to the naive clipping approach in the context of time series classification. With exactly the same experimental setting as in Section 3.4.2, we use the sparsely sampled OGLEII data set where each time series is downsampled to have 10 points. As the sparse method does not handle phase shift we compare to both the algorithm in Chapter 3 and to the same algorithm without phasing running on the universally phased data. The results is shown in Figure 4-5. Comparing the two UP methods we see that the variational approach is significantly better. With phasing the original GMT performs slightly better with large number of inducing variables, but is still significantly worse when this number is small. We believe that this difference will become more important and prominent when inputs are high dimensional and therefore harder to cover with a naive dense sample.

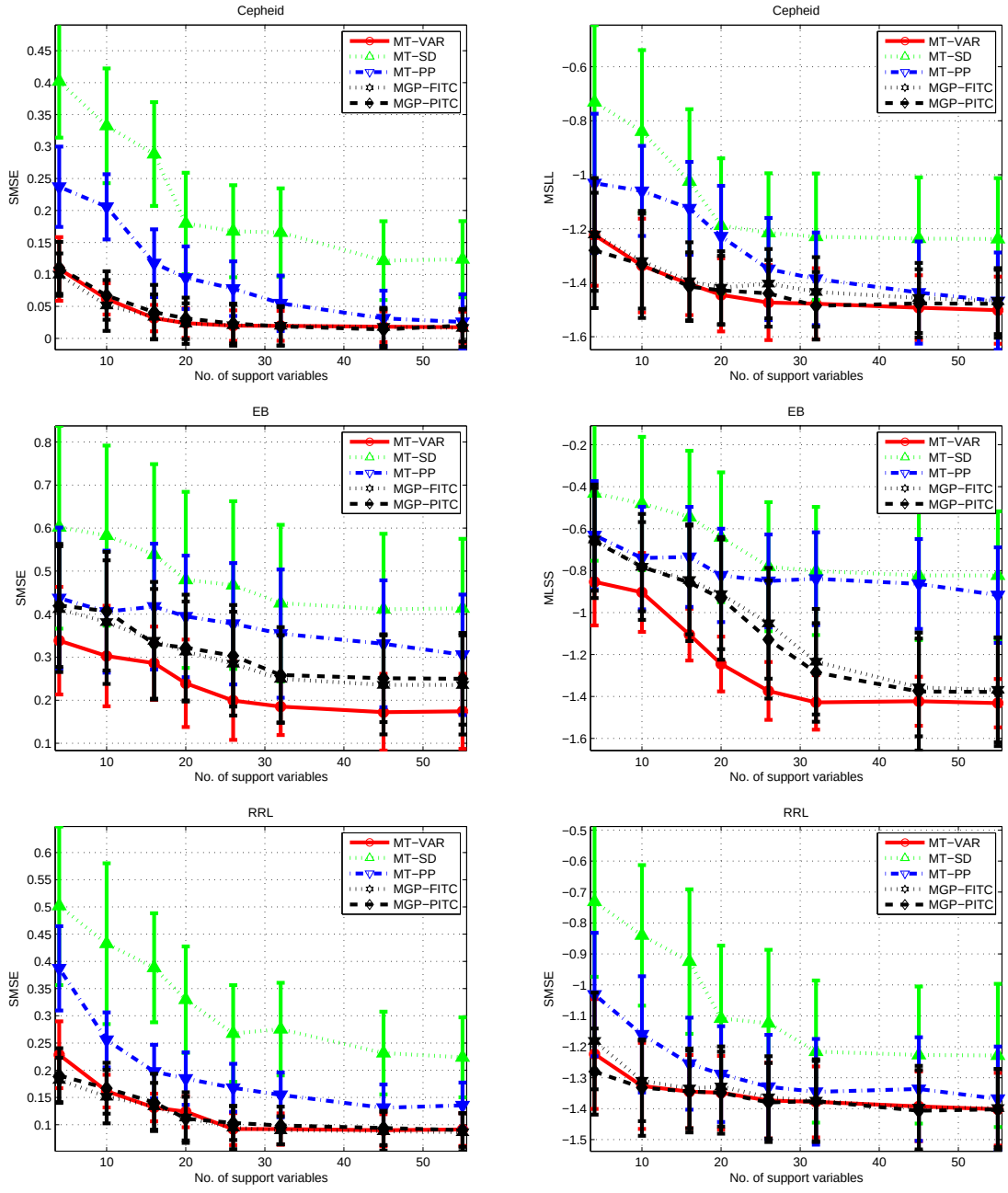

Figure 4-4: OGLEII: The average SMSE and MSLL for all the tasks are shown in the Left and Right Column. Top: Cepheid; Middle: EB; Bottom: RRL.

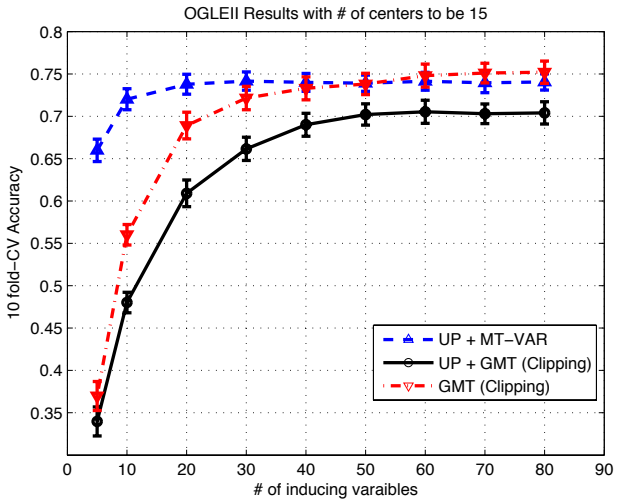

Figure 4-5: Sparsly-sampled OGLEII: Classification Results.

4.5 Related Work

Our work is related to (Titsias, 2009) particularly in terms of the form of the variational distribution of the inducing variables. However, our model is much more complex than the basic GP regression model. With the mixture model and an additional random effect per task, we must take into account the coupling of the random effect and group specific fixed-effect functions. The technical difficulty that the coupling introduces is addressed in this Chapter, yielding a generalization that is consistent with the single-task solution.

The other related thread comes from the area of GP for multi-task learning. Bonilla et al. (2008) proposed a model that learns a shared covariance matrix on features and a covariance matrix for tasks that explicitly models the dependency between tasks. They also presented techniques to speed up the inference by using the Nystrom approximation of the kernel matrix and incomplete Cholesky decomposition of the task correlation matrix. Their model, which is known as the linear coregionalization model (LCM) is subsumed by the framework of convolved multiple output Gaussian process (Álvarez and Lawrence, 2011). The work of Álvarez

and Lawrence (2011) also derives sparse solutions which are extensions of different single task sparse GP (Snelson and Ghahramani, 2006; Quiñonero-Candela and Rasmussen, 2005). Our work differs from the above models in that we allow a random effect for each individual task. As we show in the experimental section, this is important in modeling various applications. If the random effect is replaced with independent white noise, then our model is similar to LCM. To see this, from (4.38), we recognize that the posterior GP is a convex combination of K independent GPs (mean effect). However, our model is capable of prediction for newly added tasks while the models in (Bonilla et al., 2008) and (Álvarez and Lawrence, 2011) cannot. Further, the proposed model can naturally handle *heterotopic* inputs, where different tasks do not necessarily share the same inputs. In (Bonilla et al., 2008), each task is required to have same number of samples so that one can use the property of Kronecker product to derive the EM algorithm.

4.6 Conclusion

In this Chapter, we develop an efficient variational learning algorithm for the grouped mixed-effect GP for multi-task learning, which compresses the information of all tasks into an optimal set of support variables for each mean effect. Experimental evaluation demonstrates the effectiveness of the proposed method. In future, it will be interesting to derive an online sparse learning algorithm for this model. Another important direction is to investigate efficient methods for selection of inducing variables when the input is in a high dimensional space. In this case, the clipping method of (Wang et al., 2010) is clearly not feasible. The variational procedure can provide appropriate guidance, but simple gradient based optimization may not suffice.

Chapter 5

Nonparametric Bayesian Estimation of Periodic Light Curves

5.1 Introduction

Many physical phenomena exhibit periodic behavior. Discovering their period and the periodic pattern they exhibit is an important task toward understanding their behavior. In astrophysics, significant effort has been devoted to the analysis of light curves from periodic variable stars. For example, the left part of Figure 5-1 shows the magnitude of a light source over time. The periodicity of the light source is not obvious before we fold it. However, as the right part illustrates, once folded with the correct period, that is when we move the measurements at time t to time t mod T, we get convincing evidence of periodicity. The object in this figure is classified as an eclipsing binary (EB) star for the OGLE dataset used in the previous chapters. Other sources (e.g., RRL and Cepheids) show periodic variability due to processes internal to the star (Petit, 1987). In the previous chapters we used astronomy data that was already preprocessed by identifying the period and folding it. In this chapter, we address the important task of estimating the period in order to enable those algorithms to be applied on other unprocessed datasets. The problem of period estimation from noisy and irregularly sampled observations has been studied before in several disciplines. Most approaches identify the period by

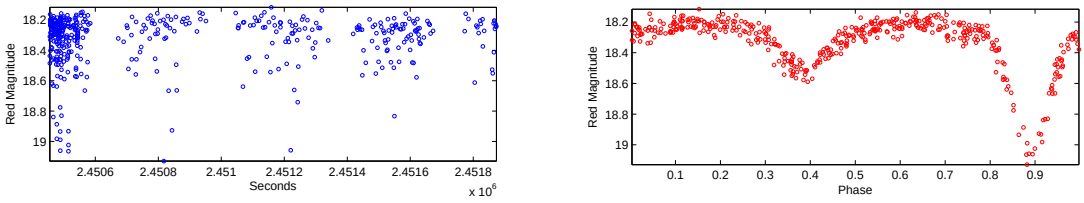

Figure 5-1: Left: brightness of an eclipsing binary (EB) star over time; Right: brightness versus phase.

some form of grid search. That is, the problem is solved by evaluating a criterion Φ at a set of trial periods $\{p\}$ and selecting the period p that yields the best value for $\Phi(p)$. The commonly-used techniques vary in the form and parametrization of Φ, the evaluation of the fit quality between model and data, the set of trial periods searched, and the complexity of the resulting procedures. Two methods we use as baselines in our study are the LS periodogram (Scargle, 1982; Reimann, 1994) and the phase dispersion minimization (PDM) (Stellingwerf, 1978), both known for their success in empirical studies. The LS method is relatively fast and is equivalent to maximum likelihood estimation under the assumption that the function has a sinusoidal shape. It therefore makes a strong assumption on the shape of the underlying function. On the other hand, PDM makes no such assumptions and is more generally applicable, but it is slower and is less often used in practice. A more extensive discussion of related work is given in Section 5.4.

This chapter makes several contributions toward solving the period estimation problem. First, we present a new model for period finding, based on Gaussian Processes (GP), that does not make strong assumptions on the shape of the periodic function. In this context, the period is a hyperparameter of the covariance function of the GP and accordingly the period estimation is cast as a model selection problem for the GP. As our experiments demonstrate, the new model leads to significantly better results compared to LS when the target function is non-sinusoidal. The model also significantly outperforms PDM when the sample size is small.

Second, we develop a new algorithm for period estimation within the GP model. In the case of period estimation the likelihood function is not a smooth function of the period parameter. This results in a difficult estimation problem which is not

well explored in the GP literature (Rasmussen and Williams, 2006). Our algorithm combines gradient optimization with grid search and incorporates several mechanisms to improve the complexity over the naive approach.

In particular we propose and evaluate: an approximation using a two level grid search, approximation using limited cyclic optimization, a method using subsampling and averaging, and a method using low-rank Cholesky approximations. An extensive experimental evaluation using artificial data identifies the most useful approximations and yields a robust algorithm for period finding.

Third, we develop a novel approach for using astrophysics knowledge, in the form of a probabilistic generative model, and incorporate it into the period estimation algorithm. In particular, we propose to employ GMT to bias the selection of periods by using it as a prior over periods or as a post-processing selection criterion choosing among periods ranked highly by the GP. The resulting algorithm is applied and evaluated on astrophysics data showing significantly improved performance over previous work.

The next section defines the period estimation problem in terms of the model selection of GP. The following three sections present our algorithm, report on experiments evaluating it and applying it to astrophysics data, and discuss related work. The final section concludes with a summary and directions for future work.

5.1.1 Problem Definition

In the case of period estimation the sample points $\boldsymbol{x}_i$ are scalars x_i representing the corresponding time points, and we denote $\boldsymbol{x} = [x_1, \ldots, x_n]^T$. The underlying function $f(\cdot)$ is periodic with unknown period p and corresponding frequency $w = 1/p$. To model the periodic aspect we use a GP with a periodic RBF covariance function,

$$\mathcal{K}_{\boldsymbol{\theta}}(x_i, x_j) = \beta \exp\left\{ -\frac{2\sin^2\left(w\pi(x_i - x_j)\right)}{\ell^2} \right\}, \tag{5.1}$$

where the set of hyperparameters of the covariance function is given by $\boldsymbol{\theta} = \{\beta, w, \ell\}$. It can be easily seen that any f generated by $\mathcal{K}_{\boldsymbol{\theta}}$ is periodic with period $T = 1/w$.

To see why this kernel gives rise to periodic functions, consider two points $x_2 = x_1 + k \cdot T + \epsilon/w\pi$, we have the joint distribution

$$\begin{bmatrix} f(x_1) \\ f(x_2) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \beta & \widetilde{\beta} \\ \widetilde{\beta} & \beta \end{bmatrix}\right),$$

where $\widetilde{\beta} = \beta \cdot \exp(-o(\epsilon^2))$. Thus, as $\epsilon \to 0$, we have

$$\mathbb{E}[f(x_2)|f(x_1)] = \frac{\widetilde{\beta}}{\beta} f(x_1) \to f(x_1)$$

and

$$\mathbf{Var}[f(x_2)|f(x_1)] = \beta - \frac{\widetilde{\beta}^2}{\beta} \to 0.$$

Figure (5-2) illustrates the role of the other two hyperparameters. We can see that β controls the magnitude of the sampled functions. At the same time, ℓ which is called characteristic length determines how sharp the variation is between two points. The plots also demonstrate that the shape of the periodic functions is highly variable. If desired, other base kernels (Rasmussen and Williams, 2006) can be used and made to be periodic in a similar manner, and as in other work it is easy to add a "trend" to the data to capture functions that are not purely periodic. In this Chapter we focus on period finding with the purely periodic kernel and leave such extensions to future work.

In our problem each star has its own period and shape and therefore each has its own set of hyperparameters. Our model, thus, assumes that the following generative process is the one producing the data. For each time series j with arbitrary sample points $\boldsymbol{x}^j = [x_1^j, \cdots, x_{N_j}^j]^T$, we first draw a zero-mean GP

$$f_j|\theta_j \sim \mathcal{GP}(0, \mathcal{K}_{\boldsymbol{\theta}_j}). \tag{5.2}$$

Then, given $\boldsymbol{x}^j$ and f_j we sample the observations

$$\boldsymbol{y}^j \sim \mathcal{N}(f_j(\boldsymbol{x}^j), \sigma_j^2 \mathbb{I}). \tag{5.3}$$

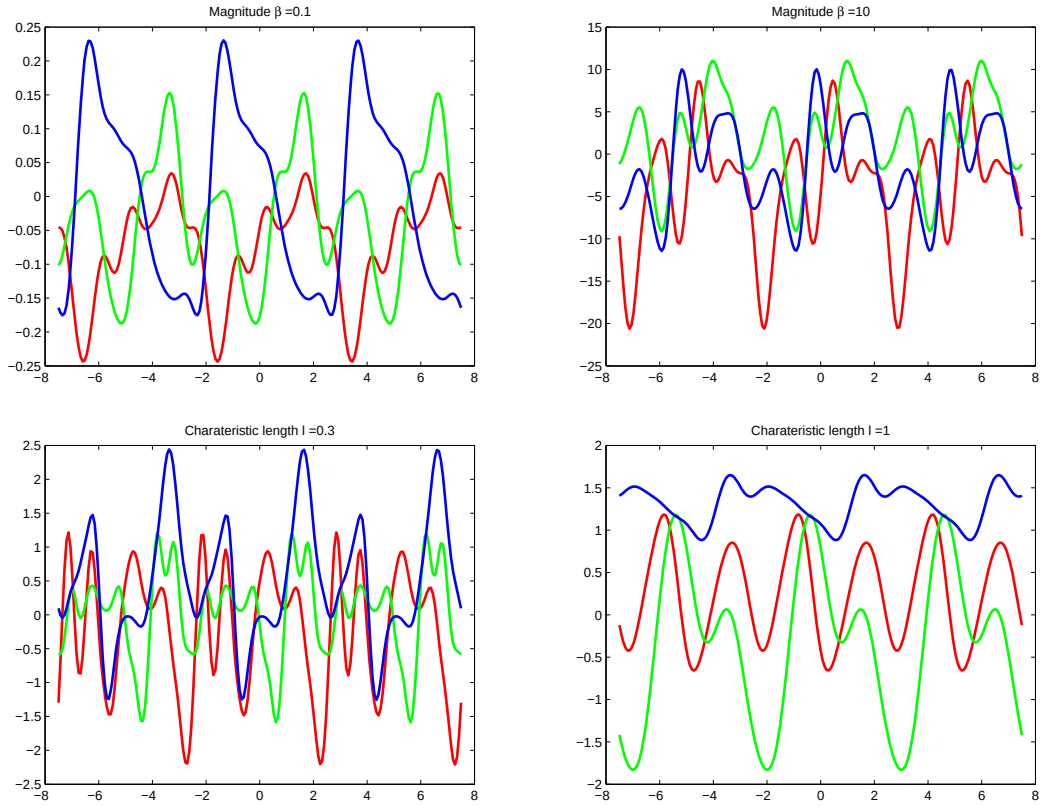

Figure 5-2: Sample functions from a GP with covariance function in (5.1) where the period is fixed to be 5, i.e. $w = 0.2$. Top row: $\beta = 0.1$ vs $\beta = 10$ while ℓ is fixed to be 0.6. Bottom row: $\ell = 0.3$ vs $\ell = 1$ with $\beta = 0.3$.

Denote the complete set of parameters by $\mathcal{M}_j = \{\boldsymbol{\theta}_j, \sigma_j^2\}$. For each time series j, the inference task is to select the correct model for the data $\{\boldsymbol{x}^j, \boldsymbol{y}^j\}$, that is, to find $\mathcal{M}_j$ that best describes the data. This is the main computational problem studied in this chapter. In the rest of this Chapter, we drop the sub-index and consider estimating $\mathcal{M}$ from $\{\boldsymbol{x}, \boldsymbol{y}\}$ as we estimate the period for each time series separately.

Before presenting the algorithm we clarify two methodological issues. First, notice that our model assumes homogeneous noise $\mathcal{N}(0, \sigma^2)$, i.e. the observation error for each x_i is the same. Experimental results on the OGLEII dataset (not shown here) show that σ^2 estimated from the data is very close to the mean of the recorded observation errors, and therefore there is no advantage in explicitly modeling the recorded observation errors. Of course, this may be different in other surveys; incorporating observation errors can be easily done by using $\sigma_{obs}^2 + \sigma^2$ in

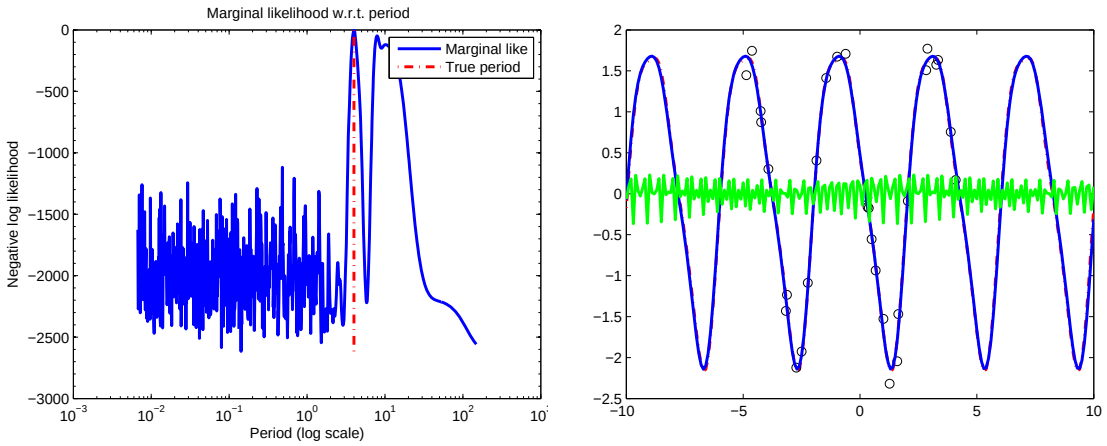

Figure 5-3: Illustration of sensitivity of the marginal likelihood. A light curve is generated using the GP model with parameters $\beta = 1$, $w = 0.25$, and $\ell = 1$. Left: The marginal likelihood function versus the period, where the dotted line indicates the true period. Right: The black circles are the observations and the dotted line (covered by the dark estimated curve) is the true function. The dark line which covers the true curve and the light line are the learned regression functions given two different starting points of w.

(5.3).

Second, as defined above our task is to find the full set of parameters $\mathcal{M}$. Therefore, our framework and induced algorithms can estimate the underlying function, f, through the posterior mean $\hat{f}$, and thus yield a solution for the regression problem – predicting the value of the function at unseen sample points. However, our main goal and interest in solving the problem is to infer the frequency w where the other parameters are less important. Therefore, a large part of the evaluation in this Chapter focuses on accuracy in identifying the frequency, although we also report results on prediction accuracy for the regression problem.

5.2 Algorithm

We start by demonstrating experimentally that gradient based methods are not sufficient for period estimation. We generate synthetic data and maximize the marginal likelihood w.r.t. $\boldsymbol{\theta} = \{\beta, w, \ell\}$ using conjugate gradients. For this experiment, 30 samples in the interval $[-10, 10]$ are generated according to the periodic

1: Initialize the parameters randomly.
2: **repeat**
3: Jointly find $\tilde{w}, \beta^*, \ell^*, \sigma^*$ that maximize (2.12) using conjugate gradients.
4: **for all** w in a coarse grid set $\mathcal{C}$ **do**
5: Calculate the marginal likelihood (2.12) or the LOO Error (2.14) using $\beta^*, \ell^*, \sigma^*$.
6: **end for**
7: Set w to the best value found in the **for** loop.
8: **until** Number of iterations reaches L_1 ($L_1 = 2$ by default)
9: Record the Top K ($K = 10$ by default) frequencies $\mathcal{W}^*$ found in the last run of **for** loop (lines 4-6).
10: **repeat**
11: Jointly find $\tilde{w}, \beta^*, \ell^*, \sigma^*$ that maximize (2.12) using conjugate gradients.
12: **for all** w in a fine grid set $\mathcal{F}$ that covers $\mathcal{W}^*$ **do**
13: Calculate the marginal likelihood (2.12) or the LOO Error (2.14) using $\beta^*, \ell^*, \sigma^*$.
14: **end for**
15: Set w to the best value found in the **for** loop.
16: **until** Number of iterations reaches L_2 ($L_2 = 2$ by default)
17: Output the frequency w^* that maximizes the marginal likelihood or minimizes the LOO Error in the last run of **for** loop (lines 11-13).

Figure 5-4: Hyperparameter Optimization Algorithm

covariance function in (5.1) with $\boldsymbol{\theta} = [1, 0.25, 1]$. Fixing β, ℓ to their correct values, the marginal likelihood w.r.t. the period $1/w$ is shown in Figure 5-3 left. The figure shows that the marginal likelihood has numerous local minima in the high frequency (small period) region that have no relation to the true period. Figure 5-3 right shows two functions with the learned parameters based on different starting points (initial values).

The function plotted in dark color estimates the true function correctly while the one in light color does not. This is not surprising because from Figure 5-3 left, we can see that there is only a small region of initial points from which the algorithm can find the correct period. We repeated this experiment using several other periodic functions with similar results. These preliminary experiments illustrate two points:

- At least for the simple data in this experiment, and when other parameters are known, the marginal likelihood function is maximized at the correct period. This shows that in principle we can find the correct period by optimizing the marginal likelihood. In practice, the region around the maximum may be very narrow, we have to deal with multiples of the correct period, and we need to account for possibly very small periods so that the problem is not so easy.

- On the other hand, the plots clearly show that it is not possible to identify the period using only gradient based search.

Therefore, as in previous work (Reimann, 1994; Hall et al., 2000), our algorithm uses grid search for the frequency. The grid used for the search must be sufficiently fine to detect the correct frequency and this implies high computational complexity. We therefore follow a two level grid search for frequency where the coarse grid must intersect the smooth region of the true maximum and the fine grid can search for the maximum itself. The two-level search significantly reduces the computational cost. Our algorithm, presented in Figure 5-4 combines this with gradient based optimization of the other parameters. There are several points that deserve further discussion, as follows:

1. In step 3, we can successfully maximize the marginal likelihood w.r.t. β, ℓ and σ^2 using the conjugate gradients method, but this approach does not work for the frequency w. The reason is that the objective function is highly sensitive w.r.t. w and the gradient is not useful for finding the global maximum. This property justifies the structure of our algorithm. This issues has been observed before and grid search (in particular using two stages) is known to be the most effective solution (Reimann, 1994; Hall et al., 2000).

2. Our algorithm uses cyclic optimization estimating w, σ, β, ℓ. That is to say, we fix other parameters σ, β, ℓ and optimize w and then optimize σ, β, ℓ when w is fixed. We keep doing this iteratively but use a small number of iterations (in our experiments, the default number of iterations is 2). A more complete algorithm

would iterate until convergence but this incurs a large computational cost. Our experiments demonstrate that a small number of iterations is sufficient.

3. In steps 3 and 11 we incorporate w into the joint optimization of the marginal likelihood. This yields better results than optimizing w.r.t. the other parameters with fixed w. This shows that the gradient of w sometimes still provides useful information locally, although the obtained optimal value $\tilde{w}$ is discarded.

4. We use an adaptive search in the frequency domain, where at the first stage we use a coarse grid and later a fine grid search is performed at the neighbors of the best frequencies previously found. By doing this, the computational cost is dramatically reduced while the accuracy of the algorithm is still guaranteed.

5. Two possible improvements to the algorithm that might appear useful are less effective than our algorithm. First, in the coarse grid search, optimizing β, ℓ and σ^2 for each w separately is too expensive because each computation of the gradient requires costly inversion of the kernel matrix. Second, one might be tempted to replace the fine grid search with a gradient based search for the optimal w. Our experiments on OGLEII (not reported here) show that this routine is inferior both in accuracy and in time complexity. This suggests that the region around the maximum is very narrow in many cases, and shows that gradient search is expensive in this problem.

Two additional approximations are introduced next, specifically targeting the coarse and fine grids respectively and using observations that are appropriate in each case.

5.2.1 Ensemble Subsampling

The coarse grid search in lines 4-6 of the algorithm needs to compute the covariance matrix w.r.t. each frequency in $\mathcal{C}$ and invert the corresponding covariance matrix, and therefore the total time complexity is $\mathcal{O}\left(|\mathcal{C}|N^3\right)$. In addition, different stars do not share the same sampling points.[1] Therefore the covariance matrix

[1] When multiple time series have the same sampling points (as might be the case with a whole field in a survey) we can store the values of the kernel matrices and their inverses (per setting of w,

and its inverse cannot be cached to be used on all stars. The computational cost is too high when the coarse grid has a large cardinality. Our observation here is that it might suffice to get an approximation of the likelihood at this stage of the algorithm, because additional fine grid search is done in the next stage.

Therefore, to reduce the time complexity, we propose an ensemble approach that combines the marginal likelihood of several subsampled times series. The idea (Protopapas et al., 2005) is that the correct period will get a high score for all sub-samples, but wrong periods that might score well on some sub-samples (and be preferred to others due to outliers) will not score well on all of them and will thus not be chosen. For the approximation, we sub-sample the original time series such that it only contains a fraction f of the original time points, repeating the process R times. The marginal likelihood score is the average over the R repetitions. Our experiments over the synthetic dataset justify using $f = 15\%$ and $R = 10$. For OGLEII we constrain this to have at least 30 points (to maintain minimal accuracy) and at most 40 points (to limit complexity). This approximation reduces the time complexity to $\mathcal{O}\left(|\mathcal{C}| \times R \times (fN)^3\right)$.

5.2.2 First Order Approximation with Low Rank Approximation

Similar to the previous case, the time complexity of fine grid search is $\mathcal{O}(|\mathcal{F}|N^3)$. In this case we can reduce the constant factor in the $\mathcal{O}(N^3)$ term. Notice that in step 13, other parameters are fixed and the grid is fine so that the marginal likelihood is a smooth function of w. Suppose we have $w_0, w_1 \in \mathcal{F}$ where $\mathcal{F}$ is the fine grid and $\Delta w = |w_0 - w_1| < \epsilon$, where ϵ is a predefined threshold. Then, given $\mathbf{K}_{w_0}$, the covariance matrix w.r.t. w_0, we can get $\mathbf{K}_{w_1}$ by its Taylor expansion as

$$\mathbf{K}_{w_1} = \mathbf{K}_{w_0} + \frac{\partial \mathbf{K}}{\partial w}(w_0)\Delta w + o(\epsilon^2). \tag{5.4}$$

β and l) and reuse these. This has the potential to significantly reduce the time complexity of the algorithm.

Denote $\widetilde{\mathbf{K}} = \frac{\partial \mathbf{K}}{\partial w}(w_0)$ where $\widetilde{\mathbf{K}}\Delta w$ can be seen as a small perturbation to $\mathbf{K}_{w_0}$. At first look, the Sherman-Morrison-Woodbury formula (Bishop, 2006) appears to be suitable for calculating the update of the inverse efficiently. Unfortunately, preliminary experiments (not shown here) indicated that this method fails due to numerical instability. Instead, we use an update for the Cholesky factors of the matrix and calculate the inverse through these. Namely, given the Cholesky decomposition of $\mathbf{K}_{w_0} = \boldsymbol{L}\boldsymbol{L}^T$ we calculate $\widetilde{\boldsymbol{L}}$ such that $\widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{L}}^T = \mathbf{K}_{w_0} + \Delta w\widetilde{\mathbf{K}} \approx \mathbf{K}_{w_1}$.

It can be easily seen that $\widetilde{\mathbf{K}}$ is a real symmetric matrix. Denote its eigendecomposition as $\widetilde{\mathbf{K}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, then it can be written as the sum of a series of rank one components,

$$\widetilde{\mathbf{K}} = \sum_{i=1}^{N} \operatorname{sgn}(\lambda_i) \left(\sqrt{|\lambda_i|}\mathbf{u}_i\right)\left(\sqrt{|\lambda_i|}\mathbf{u}_i\right)^T \tag{5.5}$$

where λ_i is the ith eigenvalue and $\mathbf{u}_i$ is the corresponding eigenvector. Furthermore, we perform a low rank approximation to $\widetilde{\mathbf{K}}$ such that

$$\widetilde{\mathbf{K}} \approx \sum_{i=1}^{M} \operatorname{sgn}(\lambda_{(i)}) \left(\sqrt{|\lambda_{(i)}|}\mathbf{u}_{(i)}\right)\left(\sqrt{|\lambda_{(i)}|}\mathbf{u}_{(i)}\right)^T \tag{5.6}$$

where $M < N$ is a predefined rank and $\lambda_{(i)}$ and $\mathbf{u}_{(i)}$ are the ith largest (in absolute value) eigenvalue and its corresponding eigenvector. Therefore we have,

$$\mathbf{K}_{w_1} \approx \widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{L}}^{\mathrm{T}} = \boldsymbol{L}\boldsymbol{L}^T + \sum_{i=1}^{M} \operatorname{sgn}(\lambda_{(i)})((\Delta w)^{1/2}\boldsymbol{\ell}_i)((\Delta w)^{1/2}\boldsymbol{\ell}_i)^T \tag{5.7}$$

where $\boldsymbol{\ell}_i = \sqrt{|\lambda_{(i)}|}\mathbf{u}_{(i)}$. Seeger (2007) shows that $\widetilde{\boldsymbol{L}}$ can be calculated from $\boldsymbol{L}$ where each rank one update can be done in $\mathcal{O}(N^2)$. Then the complexity for calculating the Cholesky factor of $\mathbf{K}_{w_1}$ becomes $\mathcal{O}(MN^2)$. Therefore, we can choose an ϵ-net $\mathcal{E}$ of the fine grid such that $\forall w \in \mathcal{F}, \sup_{v\in\mathcal{E}} |w - v| < \epsilon$, perform the exact Cholesky decomposition directly only on the ϵ-net, and use the approximation on the other frequencies. In this way we reduce the complexity from $\mathcal{O}(|\mathcal{F}|N^3)$ to $\mathcal{O}(|\mathcal{E}|N^3 + |\mathcal{F}|MN^2)$.

5.2.3 Astrophysical Input Improvements

For some cases we may have further information on the type of periodic functions one might expect. We propose to use such information to bias the selection of periods, by using it to induce a prior over periods or as a post-processing selection criterion. The details of these steps are provided in the next section.

5.3 Experiments

This section evaluates the various algorithmic ideas using synthetic and astrophysics data and then applies the algorithm to a different set of lightcurves.

5.3.1 Synthetic data

In this section, we evaluate the performance of several variants of our algorithm, study the effects of its parameters, and compare it to the two most used methods in the literature: the LS periodogram (LS) (Lomb, 1976) and phase dispersion minimization (PDM) (Stellingwerf, 1978).

The LS method (Lomb, 1976) chooses w to maximize the periodogram defined as:

$$P_{LS}(\omega) = \frac{1}{2}\left\{ \frac{[\sum y_j \cos(\eta_j)]^2}{\sum \cos^2(\eta_j)} + \frac{[\sum y_j \sin(\eta_j)]^2}{\sum \sin^2(\eta_j)} \right\}, \tag{5.8}$$

where $\eta_j = \omega(x_j - \tau)$. The phase τ (that depends on ω) is defined as the value satisfying $\tan(2\omega\tau) = \frac{\sum \sin(2\omega x_j)}{\sum \cos(2\omega x_j)}$. As shown by (Reimann, 1994), LS fits the data with a harmonic model using least-squares.

In the PDM method, the period producing the least possible scatter in the derived light curve is chosen. The score for a proposed period can be calculated by folding the light curve using the proposed period, dividing the resulting observation phases into bins, and calculating the local variance within each bin, $\sigma^2 = \frac{\sum_j (y_j - \bar{y})^2}{N-1}$, where $\bar{y}$ is the mean value within the bin and the bin has N sam-

ples. The total score is the sum of variances over all the bins. This method has no preference for a particular shape (e.g., sinusoidal) for the curve.

We generate two types of artificial data, referred to as harmonic data and GP data below. For the first, data is sampled from a simple harmonic function,

$$y \sim \mathcal{N}\left(a\sin(\omega x + \phi_1) + b\cos(\omega x + \phi_2), \sigma^2\mathbb{I}\right) \tag{5.9}$$

where $a, b \sim \text{Uniform}(0, 5)$, $\omega \sim \text{Uniform}(1, 4)$, $\phi_i \sim \mathcal{N}(0, 1)$ and the noise level σ^2 is set to be 0.1. Note that this is the model assumed by LS. For the second, data is sampled from a GP with periodic covariance function in (5.1). We generate β, ℓ uniformly in $(0, 3]$ and $(0, 3]$ respectively and the noise level σ^2 is set to be 0.1. The period is drawn from a uniform distribution between $(0.5, 2.5]$. For each type we generate data under the following configuration. We randomly sampled 50 time series each having 100 time samples in the interval $[-5, 5]$. Then the comparison is performed using sub-samples with size increasing from 10 to 100. This is repeated ten times to generate means and standard deviations in the plots.

The setting of the algorithms is as follows: In our algorithm we only use one stage grid search. For our algorithm and LS, the lowest frequency $f_{\min}$ to be examined is the inverse of the span of the input data $1/(x_{\max} - x_{\min}) = 1/T$. The highest frequency $f_{\max}$ is N/T. For the grid, the range of frequencies is broken into even segments of $1/8T$. For PDM we set the frequency range to be $[0.02, 5]$ with the frequency increments of 0.001 and the number of bins in the folded period is set to be 15.

For performance measures we consider both "accuracy" in identifying the period and the error of the regression function. For accuracy, we consider an algorithm to correctly find the period if its error is less than 1% of the true period, i.e., $|\hat{p} - p|/p \leqslant 1\%$. For the astrophysics data set, we consider the "true period" as the period identified by domain expert. Further experiments (not shown here) justify this approach by showing that the accuracies reported are not sensitive to the predefined error threshold.

The results, where our algorithm does not use the sampling and low rank approximations, are shown in Figure 5-5 and they support the following observations.

1. As expected, the top left plot shows that LS performs very well on the harmonic data and it outperforms both PDM and our algorithm. This means that if we know that the expected shape is sinusoidal, then LS is the best choice. This confirms the conclusion of other studies. For example, in the problem of detecting periodic genes from irregularly sampled gene expressions (Wentao et al., 2008; Glynn et al., 2006), the periodic time series of interest were exactly sine curves. In this case, studies showed that LS is the most effective comparing to several other statistical models.

2. On the other hand, the top right plot shows that our algorithm is significantly better than LS on the GP data showing that when the curves are non-sinusoidal the new model is indeed useful.

3. The two plots in top row together show that our algorithm performs significantly better than PDM on both types of data, especially when the number of samples is small.

4. The first two rows show the performance of the cyclic optimization procedure with 1-5 iterations. We clearly see that for these datasets there is little improvement beyond two iterations. The bottom row shows two examples of the learned regression curves using our method with different number of iterations. Although one iteration does find the correct period, the reconstruction curves are not accurate. However, here too, there is little improvement beyond two iterations. This shows that for the data tested here two iterations suffice for period estimation and for the regression problem.

5. The performance of marginal likelihood and cross validation is close, with marginal likelihood dominating on the harmonic data and doing slightly worse in GP data.

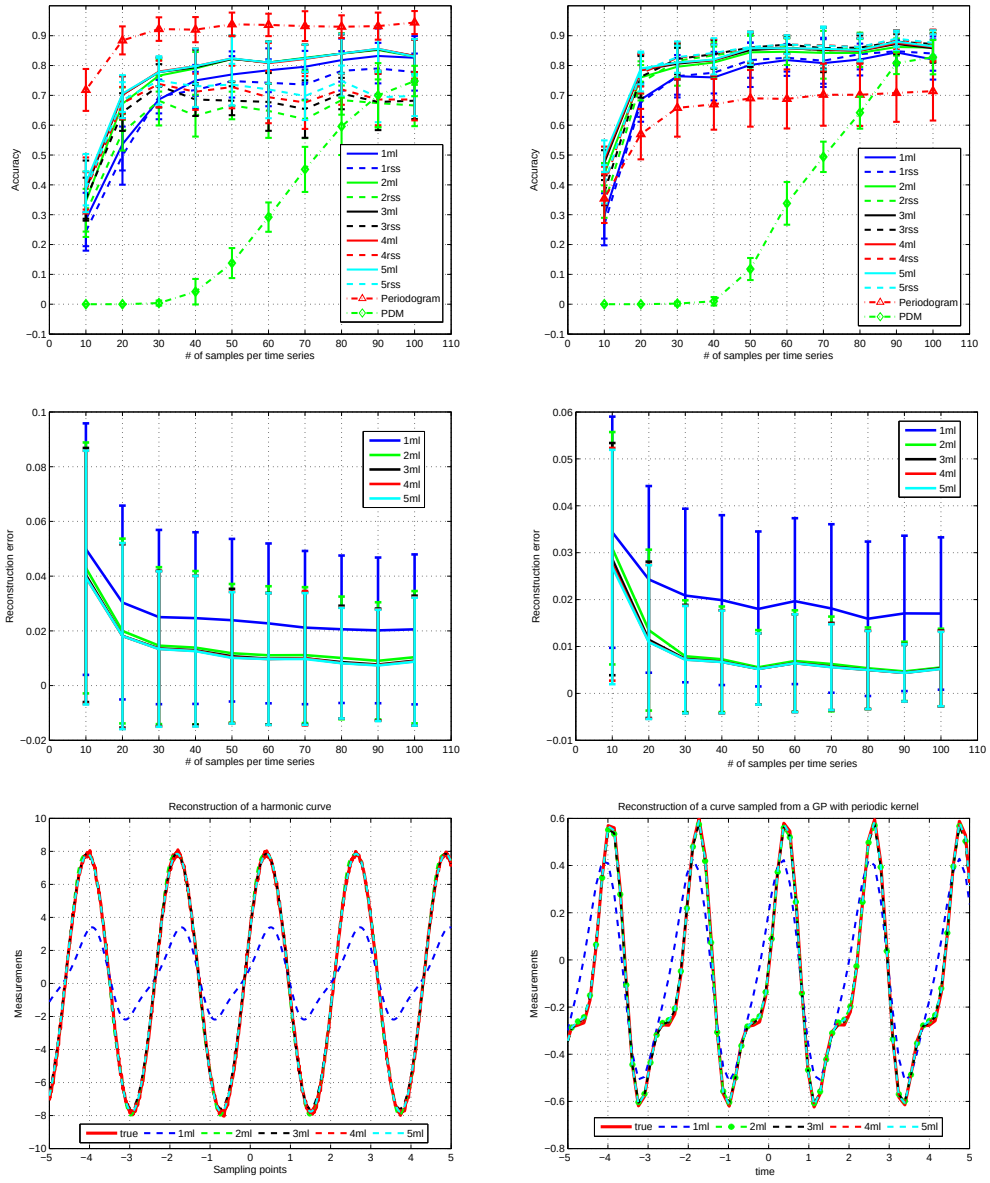

Figure 5-5: Results for harmonic data (left column) and GP data (right column). Left: Accuracy (mean and standard deviation) versus the number of samples, where solid lines marked with $n$ml represent GP with marginal likelihood where n denotes the number of iterations. The corresponding dotted lines marked $n$rss denote cross-validation results with n iterations. Middle: Reconstruction error for the regression function versus the number of samples. Right: Reconstruction curve of GP in two specific runs using maximum likelihood with different number of iterations.

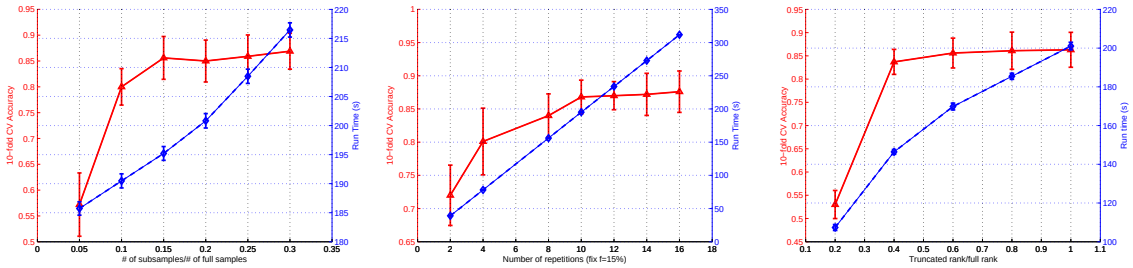

Figure 5-6: Accuracy (solid line) and Run time (dash-line) of approximation methods as a function of their parameters. Left: sub-sampling ratio (with $R = 10$). Middle: number of repetitions (with 15% sub-sampling). Right: rank in low rank approximation.

	ORIGINAL	SUBSAMPLING	SUB + LOWR
ACC	0.831 ± 0.033	$\mathbf{0.857 \pm 0.038}$	0.849 ± 0.028
S/TS	518.52 ± 121.49	197.59 ± 14.10	$\mathbf{170.75 \pm 17.93}$

Table 5.1: Comparison of GPs: Original, Subsampling and Subsampling plus low rank Cholesky update. ACC denotes accuracy and S/TS denotes the running time in seconds per time series.

We next investigate the performance of the speedup techniques. For this we use GP data under the same configuration as the previous experiments. The experiment was repeated 10 times where in each round we generate 100 lightcurves each having 100 samples but generated from different θs. For the algorithm we used two iterations for cyclic optimization and varied the subsampling size, number of repetitions and rank of the approximation. Table 5.1 shows results with our chosen parameter setting using sampling rate of 15%, 10 repetitions, approximation rank $M = \lfloor \frac{N}{2} \rfloor$ and grid search threshold $\epsilon = 0.005$. We can see that the subsampling technique saves over 60% percent of the run time while at the same time slightly increasing the accuracy. Low rank Cholesky approximation leads to an additional 15% decrease in run time, but gives slightly less good performance. Figure 5-6 plots the performance of the speedup methods under different parameter settings. The figure clearly shows that the chosen setting provides a good tradeoff in terms of performance vs. run time.

5.3.2 Astrophysics Data

In this section, we estimate the periods of unfolded astrophysics time series from the OGLEII survey.

We first explore, validate and develop our algorithm using a subset of OGLEII data and then apply the algorithm to the full OGLEII data except this development set. The OGLE subset is chosen to have 600 time series in total where each category is sampled according to its proportion in the full dataset.

Evaluating the General GP Algorithm

The setting for our algorithm is as follows: The grid search ranges are chosen to be appropriate for the application using coarse grid of $[0.02, 5]$ in the frequency domain with the increments of 0.001. The fine grid is a 0.001 neighborhood of the top frequencies each having 20 points with a step of 0.0001. We use $K = 20$ top frequencies in step 9 of the algorithm and vary the number of iterations in a cyclic optimization. When using sub-sampling, we use 15% of the original time series, but restrict sample size to be between 30 and 40 samples. This guarantees that we do not use too small a sample and that complexity is not too high. For LS we use the same configuration as in the synthetic experiment. Results are shown in Table 5.2 and they mostly confirm our conclusions from the synthetic data. In particular, the marginal likelihood (ML) is slightly better than Cross Validation (CV) and subsampling yields a small improvement. In contrast with the artificial data, more iterations do provide a small improvement in performances and 5 iterations provide the best results in this experiment. Finally, we can also see that all of the GP variants outperform LS.

Although this is an improvement over existing algorithms, accuracy of 80% is still not satisfactory. As discussed by Wachman (2009), one particularly challenging task is finding the true period of EB stars. The difficulty comes from the following two aspects. First, for a symmetric EB, the true period and half of the true period are not clearly distinguishable quantitatively. Secondly, methods that

	GP-ML	GP-CV	SGP-ML	SGP-CV	LS
1ITR ACC	0.7856	0.7769	0.7874	0.7808	0.7333
2ITR ACC	0.7892	0.7805	0.7910	0.7818	-
3ITR ACC	0.7928	0.7806	0.7964	0.7845	-
4ITR ACC	0.7946	0.7812	0.7982	0.7875	-
5ITR ACC	0.7964	0.7823	0.8000	0.7906	-

Table 5.2: Comparisons of different GPs on OGLEII subset. GP-ML and GP-CV are GP with the ML and CV criteria. SGP-ML and SGP-CV are the corresponding subsampling versions. The first column denotes the number of iterations.

are better able to identify the true period of EBs are prone to find periods that are integer multiples of single bump stars like RRLs and Cepheids. On the other hand, methods that fold RRLs and Cepheids correctly often give "half" of the true period of EBs. In particular, the low performance of LS is due to the fact that it gives a half or otherwise wrong period for most EBs.

To illustrate the results Figure 5-7 shows the periods found by LS and by GP on 4 stars. The top row shows 2 cases where the GP method finds the correct period and LS finds half the period. The bottom row shows cases where LS identifies the correct period and the GP does not. In the example on the left the GP doubles the period. In the example on the right the GP identifies a different period from LS but given the spread in the correct period the period it uncovers is not unreasonable.

Incorporating Domain Knowledge

We next show how this issue can be alleviated and the performance can be improved significantly using a learned probabilistic generative model. The methods developed are general and can be applied whenever such a model is available.

As discussed in Chapter 3, periodic stars come in families and the GMT model can learn the shapes of subgroups through the mean effect of each group. Once model parameters are learned we can calculate the likelihood of a light curve folded using a proposed period. Given the models, learned from a disjoint set of time series, for Cepheids, EBs and RRLs with parameter sets $\mathcal{M}_i, i = \{C, E, R\}$, there are two perspectives on how they can be used:

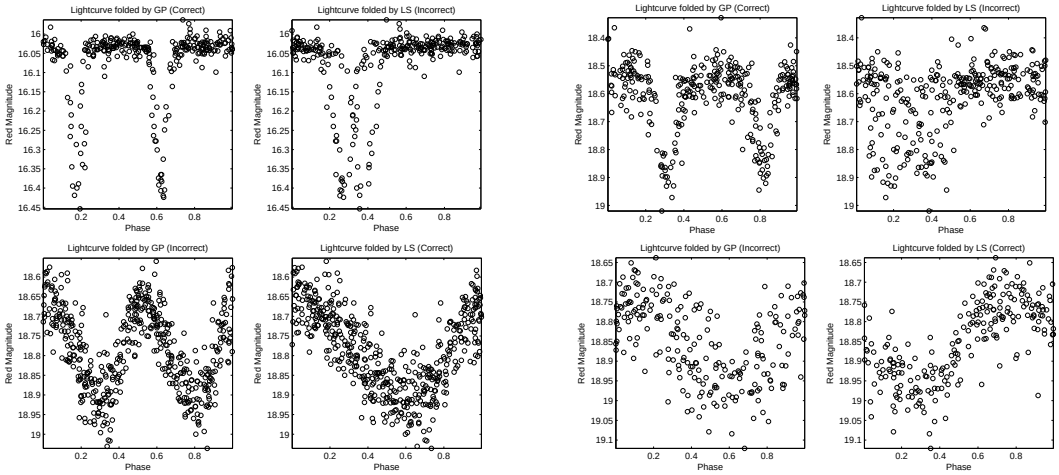

Figure 5-7: Examples of light curves where GP and LS and identify different periods and one of them is correct. Each pair shows the time series folded by GP on the left and LS on the right. The top row shows cases where LS identifies half the period. The bottom row shows cases where GP identifies double the period or a different period.

γ	0	.1	.3	.5	.7	.9	1
ACC	0.87027	0.85946	0.81802	0.81802	0.80901	0.80721	0.8

Table 5.3: Comparison of different regularization parameters on OGLEII subset using MAP.

Model as Prior: The models can be used to induce an improper prior distribution (or alternatively a penalty function) on the period p. Given period p and sample points $\boldsymbol{x}$ the prior is given by

$$\Pr(p) = \max_{i \in \{C,E,R\}} \left(\Pr(\boldsymbol{y}|\boldsymbol{x}, p; \mathcal{M}_i)\right) \tag{5.10}$$

where from the perspective of $\mathcal{M}_i$, $\boldsymbol{x}$ and corresponding points in $\boldsymbol{y}$ are interpreted as if they were sampled modulo p. Thus, combining this prior with the marginal likelihood, a Maximum A Posteriori (MAP) estimation can be obtained. Adding a regularization parameter γ to obtain a tradeoff between the marginal likelihood and the improper prior we get our criterion:

$$\begin{aligned} \log \Pr(p|\boldsymbol{x}, \boldsymbol{y}; \mathcal{M}) = \gamma \log \Pr(\boldsymbol{y}|\boldsymbol{x}, p; \mathcal{M}) \\ + (1-\gamma) \log \Pr(p) \end{aligned} \tag{5.11}$$

where $\Pr(\boldsymbol{y}|\boldsymbol{x}, p; \mathcal{M})$ is exactly as (2.12) where the period portion of $\mathcal{M}$ is fixed to be p. When using this approach with our algorithm we use (5.11) instead of (2.12) as the score function in lines 5 and 13 of the algorithm. The results for different values of γ (with subsampling and 5 iterations) are shown in Table 5.3. The results show that GMT on its own ($\gamma = 0$) is a good criterion for period finding. This is as one might expect because the OGLEII dataset includes only stars of the three types captured by GMT.

In this experiment, regularized versions do not improve the result of the GMT model. However, we believe that this will be the method of choice in other cases when the prior information is less strong. In particular, if the data includes unknown shapes that are not covered by the generative model then the prior on its own will fail. On the other hand when using (5.11), with enough data the prior will be dominated by the likelihood term and therefore the correct period can be detected. In contrast, the filter method discussed next does not have such functionality.

Model as Filter: Our second approach uses the model as a post-processing filter and it is applicable to any method that scores different periods before picking the top scoring one as its estimate. For example, suppose we are given the top K best periods $\{p_i\}, i = 1, \cdots, K$ found by LS, then we choose the one such that

$$p^* = \underset{i \in \{1,\cdots,K\}}{\operatorname{argmax}} \left(\max_{j \in \{C,E,R\}} \left[\log \Pr(\boldsymbol{y}|\boldsymbol{x}, p_i; \mathcal{M}_j) \right] \right). \tag{5.12}$$

Thus, when using the GMT as a filter, step 17 in our algorithm is changed to record the top K frequencies from the last **for** loop, evaluate each one using the GMT model likelihood, and output the top scoring frequency.

Heuristic for Variable Periodic Stars: The two approaches above are general and can be used in any problem where a model is available. For the astrophysics problem we develop another heuristic that specifically addresses the half period problem of EBs. In particular, when using the filter method, instead of choosing the top K periods, we double the selected periods, evaluate both the original and doubled

	ORIGINAL	SINGLE FILTER	FILTER
LS	0.7333	0.7243	**0.9053**
GP	0.8000	0.8829	**0.9081**
LS+GP	-	0.8811	**0.9297**

Table 5.4: Comparisons of the Accuracy of different algorithms on OGLEII subset using the GMT as a filter. SINGLE denotes without the double period heuristic.

	METHOD IN (WACHMAN, 2009)	LS-FILTER	GP-FILTER	GP-LS-FILTER
ACC	0.8680	0.8975 ± 0.04	0.8963 ± 0.03	$\mathbf{0.9243 \pm 0.03}$

Table 5.5: Comparisons of accuracies for full set of OGLEII.

periods $\{p_i, 2p_i\}$ using the GMT model, and choose the best one.

Results of experiments using the filter method with and without the domain specific heuristic are given in Table 5.4, based on the 5 iteration version of subsampling GP. The filter method significantly improves the performance of our algorithm showing its general applicability. The domain specific heuristic provides an additional improvement. For LS, the general filter method does not help but the domain specific heuristic significantly improves its performance. By analyzing the errors of both GP and LS, we found that their error regions are different. Therefore, we further propose a method that combines the two methods in the following way: pick the top K periods found by both methods and evaluate the original and doubled periods using the GMT to select the best one. As Table 5.4 shows, the combination gives an additional 2% improvement on the OGLEII subset.

Application

Finally, we apply our method using marginal likelihood with two level grid search, sub-sampling, 2 iterations, and filtering on the complete OGLEII data set minus the development OGLEII subset. Note that the parameters of the algorithm, other than domain dependent heuristics, are chosen based on our results from the artificial data. The accuracy is reported using 10-fold cross validation under the following setting: the GMT is trained using the training set and we seek to find the periods

for the stars in the test set. We compare our results to the best result from (Wachman, 2009) that used an improvement of LS, despite the fact that they filtered out 1832 difficult stars due to insufficient sampling points and noise. The results are shown in Table 5.5. We can see that our approach significantly outperforms existing methods on OGLEII.

5.4 Related Work

Period detection has been extensively studied in the literature and especially in astrophysics. The periodogram, as a tool for spectral analysis, dates back to the 19th century when Schuster applied it to the analysis of some data sets. The behavior of the periodogram in estimating frequency was discussed by Deeming (1975). The periodogram is defined as the modulus-squared of its discrete Fourier transform (Deeming, 1975). Lomb (1976) and Scargle (1982) introduced the so-called Lomb-Scargle (LS) Periodogram that was discussed above and which rates periods based on the sum-of-squares error of a sine wave at the given period. This method has been used in astrophysics (Cumming, 2004; Wachman, 2009) and has also been used in Bioinformatics (Glynn et al., 2006; Wentao et al., 2008). One can show that the LS periodogram is identical to the equation we would derive if we attempted to estimate the harmonic content of a data set at a specific frequency using the linear least-squares model (Scargle, 1982). This technique was originally named least-squares spectral analysis method Vaníček (1969). Many extensions of the LS periodogram exist in the literature (Bretthorst, 2001). Hall and Li (2006) proposed the periodogram for non-parametric regression models and discussed its statistical properties. This was later applied to the situation where the regression model is the superposition of functions with different periods (Hall, 2008).

The other main approach uses least-squares estimates, equivalent to maximum likelihood methods under Gaussian noise assumption, using different choices of periodic regression models. This approach, using finite-parameter trigonometric series of different orders, has been explored by various authors (Hartley, 1949;

Quinn and Thomson, 1991; Quinn and Fernandes, 1991; Quinn, 1999; Quinn and Hannan, 2001). Notice that if the order of the trigonometric series is high then this is very close to nonparametric methods (Hall, 2008).

Another intuition is to minimize some measure of dispersion of the data in phase space. Phase Dispersion Minimization (Stellingwerf, 1978), described above, performs a least squares fit to the mean curve defined by averaging points in bins. Lafler and Kinman (1965) described a procedure which involves trial-period folding followed by a minimization of the differences between observations of adjacent phases.

Other least squares methods use smoothing based on splines, robust splines, or variable-span smoothers. Craven and Wahba (1978) discussed the problem of smoothing periodic curve with spline functions in the regularization framework and invented the generalized cross-Validation (GCV) score to estimate the period of a variable star. Oh et al. (2004) extended it by substituting the smoothing splines with robust splines to alleviate the effects caused by outliers. Supersmoother, a variable-span smoother based on running linear smooths, is used for frequency estimation in (McDonald, 1986).

Several other approaches exist in the literature. Perhaps the most related work is (Hall et al., 2000) who studied nonparametric models for frequency estimation, including the Nadaraya-Watson estimator, and discussed their statistical properties. This was extended to perform inference for multi-period functions (Hall and Yin, 2003) and evolving periodic functions (Genton and Hall, 2007; Hall, 2008). Our work differs from (Hall et al., 2000) in three aspects: 1) the GP framework presented in this Chapter is more general in that one can plug in different periodic covariance functions for different prior assumptions; 2) we use marginal likelihood that can be interpreted to indicate how the data agrees with our prior belief; 3) we introduce mechanisms to overcome the computational complexity of period selection.

Other approaches include entropy minimization (Huijse et al., 2011), data compensated discrete Fourier transform (Ferraz-Mello, 1981), and Bayesian models

(Gregory and Loredo, 1996; Scargle, 1998). Recently, Bayesian methods have also been applied to solve the frequency estimation problem, for example Bayesian binning for Poisson-regime (Gregory and Loredo, 1996) and Bayesian blocks (Scargle, 1998). Ford et al. (2011) proposed a Bayesian extension of multi-period LS that is capable of estimating periodic functions having an additional polynomial trend. The main difference to our work is the kernel based formulation in our approach.

5.5 Conclusion

In this chapter, we introduce a nonparametric Bayesian approach for period estimation based on Gaussian process regression. We develop a model selection algorithm for GP regression that combines gradient based search and grid search, and incorporates several algorithmic improvements and approximations leading to a considerable decrease in run time. The algorithm performs significantly better than existing state of the art algorithms when the data is not sinusoidal. Further, we show how domain knowledge can be incorporated into our model as a prior or post-processing filter, and apply this idea in the astrophysics domain. Our algorithm delivers significantly higher accuracy than existing state of the art in estimating the periods of variable periodic stars.

An important direction for future work is to extend our model to develop a corresponding statistical test for periodicity, that is, to determine whether a time series is periodic. This will streamline the application of our algorithm to new astrophysics catalogs such as MACHO (Alcock et al., 1993) where both periodicity testing and period estimation are needed. Another important direction is establishing the theoretical properties of our method. Hall et al. (2000) provided the first-order properties of nonparametric estimators such that under mild regularity conditions, the estimator is consistent and asymptotically normally distributed. Our method differs in two ways: we use a GP regressor instead of Nadaraya-Watson estimator, and we choose the period that minimizes marginal likelihood rather than using a cross-validation estimate. Based on the well known connection

between kernel regression and GP regression, we conjecture that similar results exist for the proposed method.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we have studied GPs for multi-task learning under the framework of mixed-effects models. In summary, we have made the following achievements.

1. We propose a family of novel nonparametric Bayesian models for multi-task learning and show how they can be used for *time series classification, clustering and prediction* in **Chapter 3** and **Chapter 4**.

2. We propose, in **Chapter 3**, the *Shift-invariant Grouped Mixed-effects GP* (GMT) and Infinite GMT (DP-GMT) that are capable of 1) dealing with phase shift for periodic time series; 2) performing automatic model selection. We develop details for the EM algorithm for the GMT model and a Variational EM algorithm for DP-GMT optimizing the MAP estimates for the parameters of the models. The main insights in the GMT solution are in estimating the expectation for the coupled hidden variables (the cluster identities and the task specific portion of the time series) and in solving the regularized least squares problem for a set of phase-shifted observations. In addition, for the DP-GMT, we show that the variational EM algorithm can be implemented with the same complexity as the fixed order GMT without using sampling. Thus the DP-GMT provides an efficient model selection algorithm compared to alternatives such as Bayesian Information Criterion (BIC). As a special case

our algorithm yields the (Infinite) Gaussian mixture model for phase shifted time series, which may be of independent interest, and which is a generalization of the algorithms of Rebbapragada et al. (2009) and Gaffney and Smyth (2003).

3. To address the Achilles' heel (cubic time complexity) of GP based approaches, in **Chapter 4**, we propose a sparse solution for the *Grouped Mixed-effects GP* model. Specifically, we extend the approach of Titsias (2009) and develop a variational approximation that allows us to efficiently learn the shared hyperparameters and choose the sparse pseudo samples. In addition, we show how the variational approximation can be used to perform prediction efficiently once learning has been performed. Our approach is particularly useful when individual tasks have a small number of samples, different tasks do not share sampling points, and there is a large number of tasks. Our experiments, using artificial and real data, validate the approach showing that it can recover the performance of inference with the full sample, that it performs better than simple sparse approaches for multi-task GP, and that for some applications it significantly outperforms alternative sparse multi-task GP formulations (Álvarez and Lawrence, 2011).

4. Finally, we introduce the period estimation problem to the machine learning community and develop a new algorithm for period estimation with the GP model. In the case of period estimation the likelihood function is not a smooth function of the period parameter. This results in a difficult estimation problem which is not well explored in the GP literature (Rasmussen and Williams, 2006). Our algorithm combines gradient optimization with grid search and incorporates several mechanisms that include an approximation using a two level grid search, approximation using limited cyclic optimization, a method using sub-sampling and averaging, and a method using low-rank Cholesky approximations. Moreover, we develop a novel approach for using astrophysics knowledge, in the form of a probabilistic generative

model, and incorporate it into the period estimation algorithm. In particular, we propose to employ GMT to bias the selection of periods by using it as a prior over periods or as a post-processing selection criterion choosing among periods ranked highly by the GP. The resulting algorithm is applied and evaluated on astrophysics data showing significantly improved performance over previous work.

6.2 Future Work

The work in this thesis has focused on GP models where the response variable is real valued and is naturally modeled as a classical regression task. As illustrated in the introduction the GMT model is natural for many application domains including astronomy and medicine. On the other hand, in some problems, the label comes from a discrete captures count events, or has some other specific distribution. Moreover, a generalization of GMT to capture such response variables would be natural and it can provide a flexible and powerful prediction model for the application. As we discuss next this is the case, for example, in epidemiology.

Tracking and predicting the occurrences of diseases and their number is an important task for society as it can help in planning, prevention, and timely intervention. In recent years, the big-data phenomenon has become relevant for epidemiology because we now have access to historical records of hospitalizations and other medical conditions, and these can be cross-referenced with historical data about the same localities including weather events, population size and population at risk, environmental factors such as contamination, and socio-economic factors such population density, and average wealth of the population. All of these may be significant factors for the occurrence and spread of different diseases.

Considering every location as a separate task, it would be interesting to develop a prediction model to capture and predict the number of events in a location at any time and setting of local measurements. Hypothesizing that the behavior in different locations can be grouped into types we get a natural fit for the GMT

model.

Therefore, one immediate future work is to extend the proposed models to the aforementioned count regression, and more general settings where the response variable y_j is not normally distributed. This will allow us to apply the proposed model to a wider range of problems.

More precisely, we wish to propose the a novel *GP generalization of the mixture of generalized linear mixed effect models* of multi-task learning. Given M related tasks $\{\mathcal{D}^j\}$, we assume the following generative model,

1. Draw $\bar{f}_s|\mathcal{K}_0 \sim \exp\left\{-\frac{1}{2}\|\bar{f}_s\|^2_{\mathcal{H}_0}\right\}, \quad s = 1, 2, \cdots, k$
2. For the jth learning task,
   - Draw $z_j|\boldsymbol{\alpha} \sim \text{Multinomial}(\boldsymbol{\alpha})$
   - Draw $\tilde{f}^j|\mathcal{K} \sim \exp\left\{-\frac{1}{2}\|\tilde{f}^j\|^2_{\mathcal{H}}\right\}$
   - For the ith example in jth learning task,
     - Draw $y^j_i|z_j, f^j, \boldsymbol{x}^j_i \sim \Pr(y^j|f^j(\boldsymbol{x}^j_i))$, where $f^j = \bar{f}_{z_j} * \delta_{t_j} + \tilde{f}^j$.

That is, the generating process is the same as the GMT model but we allow the likelihood $\Pr(\boldsymbol{y}^j|f^j(\boldsymbol{x}^j_i))$ to be more general instead of multivariate normal. We are interested in the following two cases,

- **Classification:** In this case, the prediction is binary $\{-1, +1\}$ and we use the logistic function that gives the following likelihood,

$$\Pr(y^j = +1|f^j(\boldsymbol{x}^j_i)) = \frac{1}{1 + \exp(-f^j(\boldsymbol{x}^j_i))}.$$

- **Counting:** In this case, the prediction is a non-negative number in $\mathbb{N}^+ \cup 0$ and we use the Poisson distributiopn,

$$\begin{aligned}\Pr(y^j = p|f^j(\boldsymbol{x}^j_i)) &= \textbf{Poission}(\exp(-f^j(\boldsymbol{x}^j_i))) \\ &= \frac{\exp(-f^j(\boldsymbol{x}^j_i))^p \exp(-\exp(-f^j(\boldsymbol{x}^j_i)))}{p!}.\end{aligned}$$

In the following section, we will present some preliminary work towards this direction with the simplest case where we only have one task and no random effect, i.e. the standard GP model. We wish to extend the variational sparse solution in Chapter 4 to general likelihood functions that include the aforementioned counting and binary classification setting. Thus we provide a sparse solution for generalized GP models using Gaussian approximation as in (Opper and Archambeau, 2009) .

Sparse GP: a Variational Approach

Given data $\{\boldsymbol{x}_i, y_i\}, i = 1, \cdots, N$, we begin with the standard GP generative model,

$$f \sim \mathbf{GP}(0, \mathcal{K}_{\boldsymbol{\theta}}), \quad y_i \sim \Pr(y_i | f(\boldsymbol{x}_i)).$$

Denote $\mathbf{f} := [f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_N)]^{\mathrm{T}}$. The prior distribution of the latent function is $\Pr(\mathbf{f}) = \mathcal{N}(\mathbf{0}, \mathbf{K})$ where $\mathbf{K}$ is the kernel matrix ($N \times N$) with (i, j) entry $\mathcal{K}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. We are interested in the following three tasks

1. **Inference**: calculate the posterior distribution over the hidden variables $\mathbf{f}$.

$$\Pr(\mathbf{f}|\mathcal{Y}) \propto \prod_{i=1}^{N} \Pr(y_i|\mathbf{f}_i)) \Pr(\mathbf{f}).$$

2. **Prediction**: calculate the predictive distribution for a test point $\boldsymbol{x}^*$,

$$\Pr(y^*|\mathcal{Y}) = \int \Pr(y^*|\mathbf{f}(\boldsymbol{x}^*)) \Pr(f(\boldsymbol{x}^*)|\mathbf{f}) \Pr(\mathbf{f}|\mathcal{Y}) \, \mathrm{d}f(\boldsymbol{x}^*) \, \mathrm{d}\mathbf{f}.$$

3. **Model Selection**: find the best hyperparameter $\boldsymbol{\theta}^*$ that maximizes the marginal likelihood

$$\Pr(\mathcal{Y}|\boldsymbol{\theta}) = \int \Pr(\boldsymbol{y}|\mathbf{f}) \Pr(\mathbf{f}|\boldsymbol{\theta}) \, \mathrm{d}\mathbf{f}.$$

Notice that only when the likelihood is Gaussian, we can obtain the closed form solution. As in the regression case, standard solution for estimation or even approximate the posterior distribution cost $\mathcal{O}(N^3)$ time. Following the ideas in Chapter 4,

we use a set of m auxiliary variables $(X_m, \mathbf{f}_m)$ and wish to find the "best" auxiliary variables via variational model selection.

Let us briefly recap the approach discussed in Chapter 4. Firstly, we augment the latent variables $\mathbf{f}$ to be $\hat{\mathbf{f}} = \{\mathbf{f}, \mathbf{f}_m\}$ with prior distribution $\Pr(\mathbf{f}, \mathbf{f}_m) = \Pr(\mathbf{f}|\mathbf{f}_m)\Pr(\mathbf{f}_m)$ where the first part is conditional Gaussian and the second part comes from the GP prior $\Pr(\mathbf{f}_m) = \mathcal{N}(\mathbf{0}, \mathbf{K})$. The likelihood remains $\prod_i \Pr(y_i|\mathbf{f}_i))$. Up till now, the model remains exact.

The approximation comes with the following step. Direct inference over $\hat{\mathbf{f}}$ (i.e. calculating $\Pr(\hat{\mathbf{f}}|\mathcal{Y})$ requires cubic time. To obtain a sparse solution, we consider finding a variational approximation to the posterior distribution. The variational distribution is assumed to be $q(\mathbf{f}, \mathbf{f}_m) = \Pr(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)$ where the first part preserves the conditional Gaussian format and the second part is the one of primary interest.

We wish to find the variational distribution $q(\mathbf{f}, \mathbf{f}_m)$ that approximates the true posterior distribution $\Pr(\mathbf{f}, \mathbf{f}_m|\mathcal{Y})$ as closely as possible. To this end, we find the one that minimizes the KL divergence $\mathbf{KL}(q(\mathbf{f}, \mathbf{f}_m)||\Pr(\mathbf{f}, \mathbf{f}_m|\mathcal{Y}))$. At the same time, we are maximizing a variational lower bound (VLB) on the *true marginal likelihood*,

$$
\begin{aligned}
\log \Pr(\boldsymbol{y}) &= \log \int \Pr(\boldsymbol{y}, \mathbf{f}, \mathbf{f}_m) d\mathbf{f} d\mathbf{f}_m \\
&\geqslant \int q(\mathbf{f}, \mathbf{f}_m) \log \frac{\Pr(\boldsymbol{y}, \mathbf{f}, \mathbf{f}_m)}{q(\mathbf{f}, \mathbf{f}_m)} d\mathbf{f} d\mathbf{f}_m \\
&= \int \Pr(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m) \log \frac{\Pr(\boldsymbol{y}|\mathbf{f}) \Pr(\mathbf{f}|\mathbf{f}_m) \Pr(\mathbf{f}_m)}{\Pr(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)} d\mathbf{f} d\mathbf{f}_m \\
&= \int \phi(\mathbf{f}_m) \left\{ \int \Pr(\mathbf{f}|\mathbf{f}_m) \log \Pr(\boldsymbol{y}|\mathbf{f}) d\mathbf{f} + \log \frac{\Pr(\mathbf{f}_m)}{\phi(\mathbf{f}_m)} \right\} d\mathbf{f}_m \\
&= \int \phi(\mathbf{f}_m) \int \Pr(\mathbf{f}|\mathbf{f}_m) \log \Pr(\boldsymbol{y}|\mathbf{f}) d\mathbf{f} d\mathbf{f}_m - \mathbf{KL}(\phi(\mathbf{f}_m)||\Pr(\mathbf{f}_m)) \\
&= \int \phi(\mathbf{f}_m) \int \Pr(\mathbf{f}|\mathbf{f}_m) \log \prod_{i=1}^{n} \Pr(y_i|\mathbf{f}_i) d\mathbf{f} d\mathbf{f}_m - \mathbf{KL}(\phi(\mathbf{f}_m)||\Pr(\mathbf{f}_m)) \\
\text{VLB} &= \sum_{i=1}^{N} \mathbb{E}_{\phi(\mathbf{f}_m)} \left[\mathbb{E}_{\mathbf{f}_i|\mathbf{f}_m} (\log \Pr(y_i|\mathbf{f}_i)) \right] - \mathbf{KL}(\phi(\mathbf{f}_m)||\Pr(\mathbf{f}_m)),
\end{aligned}
$$

where

$$\Pr(\mathbf{f}_i|\mathbf{f}_m) = \mathcal{N}(v_i, \sigma_i^2) := \mathcal{N}(\mathbf{k}_{im}\mathbf{K}^{-1}\mathbf{f}_m, K_{ii} - \mathbf{k}_{im}\mathbf{K}^{-1}\mathbf{k}_{mi}).$$

Unlike Chapter 4 where the optimal variational distribution is multivariate Normal because of the Gaussian likelihood, we no longer have closed form solution for general likelihood functions. Instead, we use Gaussian approximation and assume a Gaussian form of the variational distribution $\phi(\mathbf{f}_m) = \mathcal{N}(\boldsymbol{m}, \mathbf{V})$. Thus the marginal variational distribution is obtained

$$\begin{aligned} q(\mathbf{f}_i) &= \int \Pr(\mathbf{f}, \mathbf{f}_m|\boldsymbol{y}) d\mathbf{f}_m d\{\mathbf{f}_{j\neq i}\} = \int \Pr(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m) d\mathbf{f}_m d\{\mathbf{f}_{j\neq i}\} \\ &= \mathcal{N}(\mathbf{k}_{im}\mathbf{K}^{-1}\boldsymbol{m}, K_{ii} + \mathbf{k}_{im}\mathbf{K}^{-1}(\mathbf{V} - \mathbf{K})\mathbf{K}^{-1}\mathbf{k}_{mi}). \end{aligned} \tag{6.1}$$

Therefore, we have the variational lower bound to be

$$\begin{aligned} VLB = \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{f}_i)}(\log \Pr(y_i|\mathbf{f}_i)) \\ + \frac{1}{2}(\log|\mathbf{V}\mathbf{K}^{-1}| - \mathtt{tr}(\mathbf{V}\mathbf{K}^{-1}) - \boldsymbol{m}^T\mathbf{K}^{-1}\boldsymbol{m} + N). \end{aligned} \tag{6.2}$$

Therefore, optimizing the VLB w.r.t. the parameters $(\boldsymbol{m}, \mathbf{V})$ and the hyper-parameters in $\mathbf{K}$ solves the inference and model selection problems. In the following, we will develop a concrete algorithm for the counting problem, i.e. the likelihood is Poisson distribution. In the case of Poisson regression, the log likelihood is

$$\begin{aligned} \log \Pr(y_i|\mathbf{f}_i) &= \log(\textbf{Poission}(y_i|(\exp(\mathbf{f}_i))) \\ &= y_i\mathbf{f}_i - \exp(\mathbf{f}_i) - \log y_i!. \end{aligned}$$

Considering its expectation over $q(\mathbf{f}_i)$ we can see that the first term is simple and the second term gives rise to the moment generating function of a Gaussian. Recall that the moment generating function $\mathbb{E}[e^{tX}]$ where $X \sim \mathcal{N}(\mu, \sigma^2)$ is

$\exp(t\mu + \sigma^2 t^2/2)$ (Durrett, 2004). From (6.1), we have

$$\mathbb{E}_{q(\mathbf{f}_i)}(\log \Pr(y_i|\mathbf{f}_i)) = y_i \mathbf{k}_{im}\mathbf{K}^{-1}\boldsymbol{m} - \exp\left\{\mathbf{k}_{im}\mathbf{K}^{-1}\boldsymbol{m} + \frac{1}{2}(K_{ii} + \mathbf{k}_{im}\mathbf{K}^{-1}(\mathbf{V}-\mathbf{K})\mathbf{K}^{-1}\mathbf{k}_{mi})\right\}.$$

Denote $\boldsymbol{\Phi}_i(\boldsymbol{m}, \mathbf{V}) = \mathbf{k}_{im}\mathbf{K}^{-1}\boldsymbol{m} + \frac{1}{2}(K_{ii} + \mathbf{k}_{im}\mathbf{K}^{-1}(\mathbf{V}-\mathbf{K})\mathbf{K}^{-1}\mathbf{k}_{mi})$. Putting it all together, we obtain

$$\sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{f}_i)}(\log \Pr(y_i|\mathbf{f}_i)) = \boldsymbol{y}^T K_{nm}\mathbf{K}^{-1}\boldsymbol{m} - \sum_{i=1}^{n}\exp(\boldsymbol{\Phi}_i).$$

Our current focus is the inference problem, that is, we suppose the hyper-parameters and the pesudo inputs are known and wish to find the variational parameters $(\boldsymbol{m}, \mathbf{V})$. To this end, we optimize the VLB using a gradient based method. Before moving forward, notice that

$$\frac{\partial \boldsymbol{\Phi}_i}{\partial \boldsymbol{m}} = \mathbf{K}^{-1}\mathbf{k}_{mi}, \quad \frac{\partial \boldsymbol{\Phi}_i}{\partial \mathbf{V}} = \frac{1}{2}\mathbf{K}^{-1}\mathbf{k}_{mi}\mathbf{k}_{im}\mathbf{K}^{-1} := \frac{1}{2}\mathbf{K}_i,$$

where for convenience we denote $\mathbf{K}_i = \mathbf{K}^{-1}\mathbf{k}_{mi}\mathbf{k}_{im}\mathbf{K}^{-1}$.

Therefore the gradient is

$$\begin{aligned}
\frac{\partial \text{VLB}}{\partial \boldsymbol{m}} &= -\mathbf{K}^{-1}\boldsymbol{m} + \mathbf{K}^{-1}K_{mn}\boldsymbol{y} - \mathbf{K}^{-1}\sum_{i=1}^{n}\exp(\boldsymbol{\Phi}_i)\mathbf{k}_{mi} \\
\frac{\partial \text{VLB}}{\partial \mathbf{V}} &= \frac{1}{2}(\mathbf{V}^{-1} - \mathbf{K}^{-1} - \sum_{i=1}^{n}\exp(\boldsymbol{\Phi}_i)\mathbf{K}_i) \\
&= \frac{1}{2}(\mathbf{V}^{-1} - \mathbf{K}^{-1} - \mathbf{K}^{-1}(K_{mn}\exp(\boldsymbol{\Phi})K_{nm})\mathbf{K}^{-1}).
\end{aligned}$$

Furthermore, we have the Hessian of the VLB is

$$\begin{aligned}
\frac{\partial \text{VLB}}{\partial \boldsymbol{m}\boldsymbol{m}^T} &= -\mathbf{K}^{-1} - \sum_{i=1}^{n}\exp(\boldsymbol{\Phi}_i)\mathbf{K}_i \\
\frac{\partial \text{VLB}}{\partial \mathbf{V}\mathbf{V}^T} &= -\frac{1}{2}(\mathbf{V}^{-1}\otimes\mathbf{V}^{-1}) - \frac{1}{4}\sum_{i=1}^{n}\exp(\boldsymbol{\Phi}_i)\mathbf{K}_i\otimes\mathbf{K}_i,
\end{aligned}$$

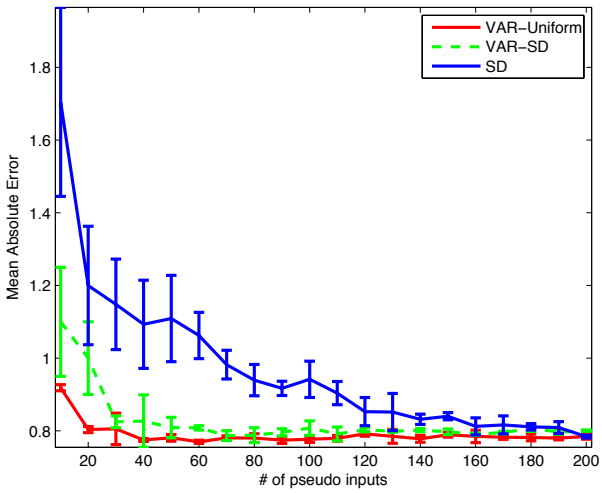

Figure 6-1: Sparse GP Poisson Regression: SD vs. Variational Approach with Evenly spaced pseudo inputs and Pseudo inputs randomly chosen from the training set (same as the SD).

where $\otimes$ denotes the Kronecker product. In the following experiment, we use the Newton method to optimize $\boldsymbol{m}$ and simple gradient descent to optimize $\mathbf{V}$. We can, in priciple use a second order method to optimize $\mathbf{V}$ using the equation above but we have found this to be numerically difficult due to the high dimension of the Hessian w.r.t. $\mathbf{V}$. We compare the proposed approach with the subset of data points (SD), which, as described in Chapter 4, only uses a portion of the training set to perform inference using the Laplace Approximation (Rasmussen and Williams, 2006). We use a synthetic data that is sampled as follows,

1. Sample a one-dimensional GP: $f \sim \mathbf{GP}(0, \mathcal{K})$ where $\mathcal{K}(s,t) = e^{-(s-t)^2/2}$.

2. Sample the counts: $y_i \sim \mathbf{Poisson}(y_i|\exp(f(\boldsymbol{x}_i))), i = 1, 2, \cdots, n$.

The hyperparameters of both approaches are fixed to be the ones that are actually used to sample the data. We take the posterior mean as the prediction and the evaluation metric is the mean absolute error, i.e. $e = \sum |\hat{y}_i - y_i|/n$. Experimental results with two types of pseudo inputs are reported in Figure 6-1. On the left hand side, we use an evenly spaced grid while a randomly subset of the original points is chosen on the right hand side. From the results, we can draw two conclusions:

1. the proposed approach converges much faster than SD;

2. even with the same pseudo inputs as the SD, the proposed approach has a significant advantage in terms of inference. Note that although the same poitns are used, the SD methods uses only the response $\{y_i\}$ at these points while the variational approach re-estimate the distribution of $\{y_i\}$ using the entire dataset and in this way it makes better use of data.

At the same time, we encountered some numerical issues when the number of pseudo inputs is large. These numerical issues, the model selection problem and the extension to the GMT model are left for future work.

Bibliography

C. Alcock et al. The MACHO Project - a Search for the Dark Matter in the Milky-Way. In B. T. Soifer, editor, *Sky Surveys. Protostars to Protogalaxies*, volume 43 of *Astronomical Society of the Pacific Conference Series*, pages 291–296, 1993.

M. Álvarez and N. Lawrence. Computationally efficient convolved multiple output Gaussian processes. *JMLR*, 12:1425–1466, 2011.

M. Álvarez, D. Luengo, M. Titsias, and N. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In *AISTATS*, 2010.

B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *The Journal of Machine Learning Research*, 4:83–99, 2003.

J. Bi, T. Xiong, S. Yu, M. Dundar, and R. Rao. An improved multi-task learning approach with applications in medical diagnosis. *European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 117–132, 2008.

S. Bickel, J. Bogojeska, T. Lengauer, and T. Scheffer. Multi-task learning for HIV therapy screening. In *Proceedings of the 25th international conference on Machine learning*, pages 56–63. ACM, 2008.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, 2006.

D. Blei and M. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144, 2006.

E. Bonilla, K. Chai, and C. Williams. Multi-task Gaussian process prediction. *NIPS*, 20:153–160, 2008.

G. Bretthorst. Generalizing the Lomb-Scargle periodogram-the nonsinusoidal case. In *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, volume 568, pages 246–251, 2001.

R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

W. Chu, Z. Ghahramani, F. Falciani, and D. L. Wild. Biomarker discovery in microarray gene expression data with gaussian processes. *Bioinformatics*, 21(16): 3385–3393, 2005.

D. Chudova, S. J. Gaffney, E. Mjolsness, and P. Smyth. Translation-invariant mixture models for curve clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 79–88. ACM, 2003.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

P. Craven and G. Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403, 1978.

A. Cumming. Detectability of extrasolar planets in radial velocity surveys. *Monthly Notices of the Royal Astronomical Society*, 354(4):1165–1176, 2004.

T. Deeming. Fourier analysis with unequally-spaced data. *Astrophysics and Space Science*, 36(1):137–158, 1975.

E. Demidenko. *Mixed models: theory and applications*. Wiley-Interscience, 2005.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B39:1–38, 1977.

P. Denti, A. Bertoldo, P. Vicini, and C. Cobelli. Ivgtt glucose minimal model covariate selection by nonlinear mixed-effects approach. *American Journal of Physiology-Endocrinology And Metabolism*, 298(5):E950, 2010.

H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment archive*, 1(2):1542–1552, 2008.

F. Dinuzzo, G. Pillonetto, and G. De Nicolao. Client-server multi-task learning from distributed datasets. *Arxiv preprint arXiv:0812.4235*, 2008.

R. Durrett. *Probability: Theory and Examples*. Duxbury Press, 2004.

T. Evgeniou, C. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(1):615–637, 2006.

L. Faccioli, C. Alcock, K. Cook, G. E. Prochter, P. Protopapas, and D. Syphers. Eclipsing Binary Stars in the Large and Small Magellanic Clouds from the MACHO Project: The Sample. *Astronomy Journal*, 134:1963–1993, 2007.

S. Ferraz-Mello. Estimation of periods from unequally spaced observations. *The Astronomical Journal*, 86:619, 1981. ISSN 0004-6256.

E. Ford, A. Moorhead, and D. Veras. A bayesian surrogate model for rapid time series analysis and application to exoplanet observations. *Bayesian Analysis*, 6(3): 475–500, 2011.

S. J. Gaffney. *Probabilistic curve-aligned clustering and prediction with regression mixture models*. PhD thesis, University of California, Irvine, 2004.

S. J. Gaffney and P. Smyth. Trajectory clustering with mixtures of regression models. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 63–72. ACM, 1999.

S. J. Gaffney and P. Smyth. Curve clustering with random effects regression mixtures. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

S. J. Gaffney and P. Smyth. Joint probabilistic curve clustering and alignment. *Advances in neural information processing systems*, 17:473–480, 2005.

A. Gelman. *Bayesian data analysis*. CRC press, 2004.

M. Genton and P. Hall. Statistical inference for evolving periodic functions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):643–657, 2007.

E. Glynn, J. Chen, and A. Mushegian. Detecting periodic patterns in unevenly spaced gene expression time series using Lomb–Scargle periodograms. *Bioinformatics*, 22(3):310, 2006.

P. Gregory and T. Loredo. Bayesian periodic signal detection: Analysis of ROSAT observations of PSR 0540-693. *The Astrophysical Journal*, 473:1059, 1996.

P. Hall. Nonparametric Methods for Estimating Periodic Functions, with Applications in Astronomy. *COMPSTAT 2008*, pages 3–18, 2008.

P. Hall and M. Li. Using the periodogram to estimate period in nonparametric regression. *Biometrika*, 93(2):411, 2006.

P. Hall and J. Yin. Nonparametric methods for deconvolving multiperiodic functions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65 (4):869–886, 2003.

P. Hall, J. Reimann, and J. Rice. Nonparametric estimation of a periodic function. *Biometrika*, 87(3):545, 2000.

H. Hartley. Tests of significance in harmonic analysis. *Biometrika*, 36(1-2):194, 1949.

K. W. Hodapp et al. Design of the Pan-STARRS telescopes. *Astronomische Nachrichten*, 325:636–642, 2004.

P. Huijse, P. A. Estevez, P. Zegers, J. C. Principe, and P. Protopapas. Period Estimation in Astronomical Time Series Using Slotted Correntropy. *IEEE Signal Processing Letters*, 18:371–374, June 2011. doi: 10.1109/LSP.2011.2141987.

H. Ishwaran and L. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001. ISSN 0162-1459.

E. Jackson, M. Davy, A. Doucet, and W. Fitzgerald. Bayesian unsupervised signal classification by dirichlet process mixtures of gaussian processes. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, volume 3, 2007.

T. Joachims. *Learning to classify text using support vector machines: Methods, theory and algorithms*. Kluwer Academic Publishers, 2002.

T. Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384. ACM, 2005.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37, November 1999.

S. Kim and P. Smyth. Segmental hidden Markov models with random effects for waveform modeling. *Journal of Machine Learning Research*, 7:969, 2006.

J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard. Gaussian process model based predictive control. In *American Control Conference, 2004. Proceedings of the 2004*, volume 3, pages 2214–2219. IEEE, 2004.

J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In *Proceedings of the twenty-first international conference on Machine learning*, page 64. ACM, 2004.

J. Lafler and T. Kinman. An RR Lyrae Star Survey with Ihe Lick 20-INCH Astrograph II. The Calculation of RR Lyrae Periods by Electronic Computer. *The Astrophysical Journal Supplement Series*, 11:216, 1965.

N. Lomb. Least-squares frequency analysis of unequally spaced data. *Astrophysics and space science*, 39(2):447–462, 1976.

Z. Lu, T. Leen, Y. Huang, and D. Erdogmus. A reproducing kernel Hilbert space framework for pairwise time series distances. In *Proceedings of the 25th interna-*

tional conference on Machine learning, pages 624–631. ACM New York, NY, USA, 2008.

J. McDonald. Periodic smoothing of time series. *SIAM Journal on Scientific and Statistical Computing*, 7:665, 1986.

C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17(1):177–204, 2005.

T. M. Mitchell. Machine learning. *Burr Ridge, IL: McGraw Hill*, 45, 1997.

J. J. Murillo-Fuentes, S. Caro, and F. Pérez-Cruz. Gaussian processes for multiuser detection in cdma receivers. *Advances in Neural Information Processing Systems*, 18:939, 2006.

R. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000. ISSN 1061-8600.

H.-S. Oh, D. Nychka, T. Brown, and P. Charbonneau. Period analysis of variable stars by robust smoothing. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 53(1):15–30, 2004.

M. Opper and C. Archambeau. The variational gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

S. Osowski, L. Hoai, and T. Markiewicz. Support vector machine-based expert system for reliable heartbeat recognition. *IEEE Transactions on Biomedical Engineering*, 51(4):582–589, 2004.

S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

M. Petit. *Variable stars*. Chichester, England and New York, John Wiley and Sons, 1987, 268 p. Translation., 1987.

G. Pillonetto, G. De Nicolao, M. Chierici, and C. Cobelli. Fast algorithms for non-parametric population modeling of large data sets. *Automatica*, 45(1):173–179, 2009. ISSN 0005-1098.

G. Pillonetto, F. Dinuzzo, and G. De Nicolao. Bayesian Online Multitask Learning of Gaussian Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):193–205, 2010.

J. C. Pinheiro and D. M. Bates. *Mixed effects models in S and S-PLUS*. Springer Verlag, 2000.

R. J. Povinelli, M. T. Johnson, A. C. Lindgren, and J. Ye. Time series classification using Gaussian mixture models of reconstructed phase spaces. *IEEE Transactions on Knowledge and Data Engineering*, 16(6):779–783, 2004.

P. Protopapas, R. Jimenez, and C. Alcock. Fast identification of transits from light-curves. *Monthly Notices of the Royal Astronomical Society*, 362(2):460–468, 2005. ISSN 1365-2966.

P. Protopapas, J. M. Giammarco, L. Faccioli, M. F. Struble, R. Dave, and C. Alcock. Finding outlier light curves in catalogues of periodic variable stars. *Monthly Notices of the Royal Astronomical Society*, 369:677–696, 2006.

B. Quinn. A fast efficient technique for the estimation of frequency: interpretation and generalisation. *Biometrika*, 86(1):213, 1999.

B. Quinn and J. Fernandes. A fast efficient technique for the estimation of frequency. *Biometrika*, 78(3):489, 1991.

B. Quinn and E. Hannan. *The estimation and tracking of frequency*. Cambridge Univ Pr, 2001.

B. Quinn and P. Thomson. Estimating the frequency of a periodic function. *Biometrika*, 78(1):65, 1991.

J. Quiñonero-Candela and C. Rasmussen. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

C. Rasmussen and H. Nickisch. Gaussian Processes for Machine Learning (GPML) Toolbox. *Journal of Machine Learning Research*, 11:3011–3015, 2010. ISSN 1533-7928.

C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. In *Advances in neural information processing systems 14: proceedings of the 2001 conference*, pages 881–888. MIT Press, 2002.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning* . The MIT Press, 2006.

U. Rebbapragada, P. Protopapas, C. E. Brodley, and C. Alcock. Finding anomalous periodic time series. *Machine Learning*, 74(3):281–313, 2009.

J. Reimann. *Frequency estimation using unequally-spaced astronomical data*. PhD thesis, UC Berkeley, 1994.

J. Scargle. Studies in astronomical time series analysis. II-Statistical aspects of spectral analysis of unevenly spaced data. *The Astrophysical Journal*, 263:835–853, 1982.

J. Scargle. Studies in astronomical time series analysis. V. Bayesian blocks, a new method to analyze structure in photon counting data. *The Astrophysical Journal*, 504:405, 1998.

B. Scholkopf and A. J. Smola. *Learning with kernels*. MIT Press, 2002.

A. Schwaighofer, V. Tresp, and K. Yu. Learning Gaussian process kernels via hierarchical Bayes. *Advances in Neural Information Processing Systems*, 17:1209–1216, 2005.

C. Seeger, M. Williams and N. Lawrence. Fast forward selection to speed up sparse gaussian process regression. In *AISTATS 9*. 2003.

M. Seeger. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14(2):69–106, 2004.

M. Seeger. Low rank updates for the Cholesky decomposition. *University of California at Berkeley, Tech. Rep*, 2007.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2): 639–650, 1994.

M. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *NIPS 18*, pages 1257–1264. 2006.

I. Soszynski, A. Udalski, and M. Szymanski. The Optical Gravitational Lensing Experiment. Catalog of RR Lyr Stars in the Large Magellanic Cloud ?? *Acta Astronomica*, 53:93–116, 2003.

B. M. Starr et al. LSST Instrument Concept. In J. A. Tyson and S. Wolff, editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 4836 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 228–239, Dec. 2002.

E. Stein and R. Shakarchi. *Real analysis: measure theory, integration, and Hilbert spaces*. Princeton University Press, 2005.

R. Stellingwerf. Period determination using phase dispersion minimization. *The Astrophysical Journal*, 224:953–960, 1978.

Y. W. Teh. Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer, 2010.

M. Titsias. Variational learning of inducing variables in sparse gaussian processes. *AISTATS*, 2009.

V. Tresp. Mixtures of Gaussian processes. *Advances in Neural Information Processing Systems*, pages 654–660, 2001.

A. Udalski, M. Szymanski, M. Kubiak, G. Pietrzynski, P. Wozniak, and Z. Zebrun. Optical gravitational lensing experiment. photometry of the macho-smc-1 microlensing candidate. *Acta Astronomica*, 47:431–436, 1997.

P. Vaníček. Approximate spectral analysis by least-squares fit. *Astrophysics and Space Science*, 4(4):387–391, 1969.

P. Vicini and C. Cobelli. The iterative two-stage population approach to ivgtt minimal modeling: improved precision with reduced sampling. *American Journal of Physiology-Endocrinology and Metabolism*, 280(1):E179, 2001.

G. Wachman. *Kernel Methods and Their Application to Structured Data*. PhD thesis, Tufts University, 2009.

G. Wachman, R. Khardon, P. Protopapas, and C. Alcock. Kernels for Periodic Time Series Arising in Astronomy. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, pages 489–505. Springer, 2009.

M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008. ISSN 1935-8237.

Y. Wang and R. Khardon. Nonparametric bayesian mixed-effect model: a sparse gaussian process approach. *arXiv preprint arXiv:1211.6653*, 2012a.

Y. Wang and R. Khardon. Sparse gaussian processes for multi-task learning. *Machine Learning and Knowledge Discovery in Databases*, pages 711–727, 2012b.

Y. Wang, R. Khardon, and P. Protopapas. Shift-Invariant Grouped Multi-task Learning for Gaussian Processes. *Machine Learning and Knowledge Discovery in Databases*, pages 418–434, 2010.

Y. Wang, R. Khardon, and P. Protopapas. Infinite shift-invariant grouped multi-task learning for gaussian processes. *arXiv preprint arXiv:1203.0970*, 2011.

Y. Wang, R. Khardon, and P. Protopapas. Nonparametric bayesian estimation of periodic light curves. *The Astrophysical Journal*, 756(1):67, 2012.

L. Wei and E. Keogh. Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 748–753. ACM New York, NY, USA, 2006.

Z. Wentao, A. Kwadwo, S. Erchin, et al. Detecting periodic genes from irregularly sampled gene expressions: a comparison study. *EURASIP Journal on Bioinformatics and Systems Biology*, 2008, 2008. ISSN 1687-4145.

Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 8:63, 2007a.

Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007b.

K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine learning*, pages 1012–1019. ACM, 2005.