Ville Tolvanen

Gaussian Processes with Monotonicity Constraint for Big Data

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 19.5.2014

Thesis supervisor:

Prof. Jouko Lampinen

Thesis advisor:

D.Sc. (Tech.) Aki Vehtari

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Ville Tolvanen		
Title: Gaussian Processes with Monotonicity Constraint for Big Data		
Date: 19.5.2014	Language: English	Number of pages: 7+58
Department of Biomedical Engineering and Computational Science		
Professorship: Computational and Cognitive Biosciences		Code: Becs-114
Supervisor: Prof. Jouko Lampinen		
Advisor: D.Sc. (Tech.) Aki Vehtari		

In this thesis, we combine recent advances in monotonicity constraints for Gaussian processes with Big Data inference of Gaussian Proceses. The new variational inference based method is developed and experimented on several simulated and real world data sets by comparing the predictive performance to Expectation Propagation and Markov chain Monte Carlo methods.

The results indicate that the new method produces good results and can be used when the data sets get so large that the computationally demanding methods cannot be used.

Keywords: Gaussian processes, monotonicity, Big data, variational inference

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Ville Tolvanen

Työn nimi: Gaussiset prosessit monotonisuusrajoituksella suurille aineistoille

Päivämäärä: 19.5.2014 Kieli: Englanti Sivumäärä: 7+58

Lääketieteellisen tekniikan ja laskennallisen tieteen laitos BECS

Professuuri: Laskennallinen ja kognitiivinen biotiede Koodi: Becs-114

Valvoja: Prof. Jouko Lampineni

Ohjaaja: TkT. Aki Vehtari

Tämän työn tarkoitus on kehittää menetelmä monotonisuusrajoitettujen Gaussisten Prosessien käyttämiseksi suurille aineistoille. Variaatiolaskentaan perustuvaa menetelmää testataan usealla simuloidulla ja oikealla aineistolla. Uuden menetelmän prediktiivistä kykyä verrataan expectation propagation menetelmään, sekä Markov chain Monte Carlo menetelmiin.
Työssä saatujen tulosten perusteella voidaan päätellä, että uusi menetelmä toimii ja sitä voidaan käyttää, kun aineistot kasvavat liian suuriksi laskennallisesti raskaille menetelmille.

Avainsanat: Gaussiset prosessit, monotonisuus, suuret aineistot, variaatiolaskenta

Preface

This work was carried out in the Department of Biomedical Engineering and Computational Science at the Aalto University School of Science.

I would like to express my deepest gratitude for my instructor and mentor Aki Vehtari for these past few years. I would also like to express my thanks to Professor Jouko Lampinen for supervising this thesis. Additionally, I would like to thank all the people in the Bayes group who have helped me during these years.

Otaniemi, 19.5.2014

Ville A. Tolvanen

Contents

Symbols and abbreviations

Symbols

$\mathrm{N}(\cdot \mid \mu, \sigma^2)$	Normal or Gaussian distribution with mean parameter μ and variance σ^2
$\mathbf{x}$	Input vector
X	Availabale input data consisting of several input vectors $\mathbf{x}_i$
f^*	Unknown function value for some input $\mathbf{x}^*$
$p(\mathbf{f} \mid X)$	Prior distribution of $\mathbf{f}$ given X
$p(\mathbf{y} \mid \mathbf{f})$	Likelihood function of $\mathbf{y}$
$p(\mathbf{f} \mid X, \mathbf{y})$	Posterior distribution of $\mathbf{f}$ given available data X and $\mathbf{y}$
Z or $p(\mathbf{y} \mid X)$	Marginal likelihood or evidence of the data
$p(y^* \mid \mathbf{x}^*, X, \mathbf{y})$	Predictive distribution of the unknown value y^* at location $\mathbf{x}^*$, given the data X and $\mathbf{y}$
K or $K(X, X)$	Covariance matrix between function values at data points X
K_* or $K(X, X_*)$	Covariance matrix between function values at data points X and function values at data points X_*
K_{**} or $K(X_*, X_*)$	Covariance matrix between function values at data points X_*
K^T	Matrix transpose of K
K^{-1}	Matrix inverse of K
Σ	Covariance matrix
$\mathrm{diag}(e_1, e_2, \ldots, e_n)$	Diagonal matrix with the diagonal $e_1, e_2, \ldots, e_n$.
I	Identrity matrix, $I = \mathrm{diag}(1, 1, \ldots, 1)$
$y \sim p(y)$	y is distributed according to $p(y)$
$p(y) \propto q(y)$	$p(y)$ is proportionally equivalent to $q(y)$, i.e. $p(y) = Cq(y)$ with some constant C that does not depend on y

Operators

$\frac{\partial}{\partial x}$	Partial derivative with respect to x
∇ or $\nabla_{\mathbf{f}}$	Gradient (vector of first derivatives) with respect to vector values $\mathbf{f}$
$\mathbb{E}_{q(f)}[h(f)]$ or $\mathbb{E}_q[h(f)]$	Expected value of $h(f)$ over the distribution $q(f)$: $\int h(f)q(f)df$

Abbreviations

GP	Gaussian process
EP	Expectation propagation
CV	Cross-validation
MAP	Maximum a posteriori
i.i.d.	Independent and identically distributed
KL	Kullback-Leibler divergence measure
SVI	Stochastic variational inference
FIC	Fully independent conditional approximation for Gaussian processes

1 Introduction

With the increasing amount of data being collected from various sources, standard statistical methods for doing inference are becoming too slow and computationally expensive. New fast algorithms and methods are needed for handling these big data sets or *Big Data*. The use of Big Data has become more and more common recently as the collection of data has increased.

Sometimes the user of the statistical methods has knowledge that the observable quantities can be monotonic with respect to some explanatory variables. This monotonicity assumption means that when the value of some quantitative explanatory variable increases, the expectation of the observable quantity always increases (monotonically increasing) or decreases (monotonically decreasing). Consider a case where the death rate for some disease is measured given how old the patient is. It is natural to assume that the death rate might be higher for patients that are older. Incorporating this kind of monotonicity information to the statistical model can often be a challenge or even impossible.

In this thesis, we develop methods and investigate how to efficiently combine the monotonicity assumption with large data sets. The methods are developed for Gaussian processes which can be considered as Bayesian nonparametric models. The developed methods combine recent advances in Gaussian process inference for large data sets and monotonicity constrainments for the Gaussian processes. The methods are experimented on several simulated and real world data sets.

The thesis is structured as follows: in section 2 we go over the basics of Bayesian inference. The appropriate terminology is introduced and explained to ease the understanding of the following sections.

In section 3 we introduce the main model of the thesis, Gaussian process. The basic inference for Gaussian processes is explained. We introduce some standard approximation tools used in Gaussian process framework. Furthermore, we explain how the monotonicity of the function values can be achieved with Gaussian processes.

In section 4 the main contributions and work of this thesis are explained. We start by introducing the first stepping stone towards Big Data Gaussian processes: Sparse Gaussian processes. We explain the variational learning and how it can be used to do fast inference in the large data sets. We continue by explaining how the variational learning can be combined with Gaussian process framework to enable Big Data inference for Gaussian processes. We end the section by proposing how to combine variational learning with the monotonicity constraints of Gaussian processes.

Section 5 explains how the parameters of the Gaussian process are chosen. We go over different ways to either choose the parameters or integrate over them. Section 6 introduces the data sets we use in this thesis, which models we compare and how we assess or compare the different models. In section 7 we go over and analyze the results. We end this thesis by discussing the properties of the introduced models in section 8.

2 Bayesian modelling

Bayesian modelling or Bayesian statistics refers to inference about unknown parameters with the use of probability models and data (Gelman et al., 2013). The defining aspect of the Bayesian modelling is the use of probability distributions of unknown quantities for expressing the uncertainty. The origin of Bayesian modelling can be traced as far back as the year 1763 (Bayes, 1763). The fundamental cornerstone of all Bayesian modelling is *Bayes' Theorem*:

$$p(\theta \mid x) = \frac{p(\theta)p(x \mid \theta)}{p(x)} = \frac{p(\theta)p(x \mid \theta)}{\int p(\theta)p(x \mid \theta)d\theta}. \tag{1}$$

Bayes Theorem is used to compute the conditional distribution of θ given x. What makes the equation (1) special is not the technical details but rather the wide applicability of the Bayes' Theorem in statistical modelling and inference.

What separates the Bayesian modelling from the *frequentistic* framework is the use of probability distributions for expressing the uncertainty of variables, rather than assuming that the variables are fixed but unknown. The a priori uncertainty assumptions on variable θ are characterized by the prior distribution $p(\theta)$. The prior distribution represents assumptions on variable θ before seeing any data. This can mean, for example, specific knowledge that the statistician doing the inference has, or some kind of universal prior distribution (*i.e.* no knowledge). The likelihood function of the parameters θ, $p(x \mid \theta)$ or sometimes denoted as $L(\theta)$ or $L(\theta \mid x)$, quantifies the likelihood of the parameter value θ after observing the data x. The likelihood function can be derived from the observation model of the data, conditioned on the observed data. However, this map is not bijective as the observation model for the data cannot be inferred from the likelihood function. This is because there can be several observation models that produce the same likelihood functions for the parameters respectively. Combining the prior knowledge of the parameter $p(\theta)$ and the information from the data, $p(x \mid \theta)$, produces the posterior probability of the parameter $p(\theta \mid x)$ which quantifies the best knowledge we have on the parameter: prior knowledge combined with the data.

The uncertainty on the parameter values is taken into account by integrating over the probability distribution representing the uncertainty. Consider for example the posterior distribution in (1) where we compute the posterior distribution of some parameter θ given data x. Assuming that after the posterior distribution has been computed, we want to estimate how some new data, x^* behaves, given the old data x with parameters θ. In Bayesian inference, the *posterior predictive distribution* of x^* is computed by averaging the predictive distribution of the new data, $p(x^* \mid \theta, x)$, over the posterior distribution of the parameters θ:

$$\begin{aligned} p(x^* \mid x) &= \int p(x^*, \theta \mid x)d\theta, \\ &= \int p(x^* \mid \theta, x)p(\theta \mid x)d\theta. \end{aligned}$$

It is possible to avoid the full Bayesian analysis of integrating over parameters by using point estimates for the posterior distribution. If we replace the full posterior

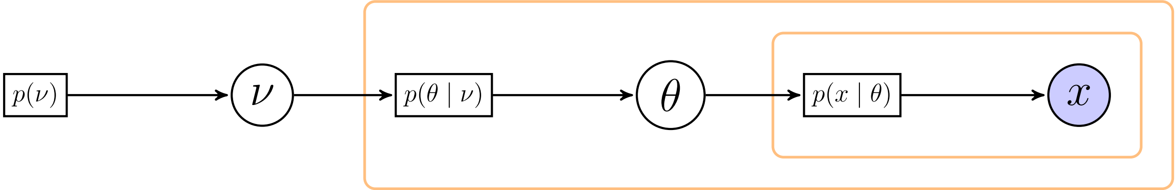


Figure 1: The graphical model of the hierarchical model structure. Circles represent the variables and rectangles represent the relations. The blue color indicates that the variable is observed. The orange rectangles emphasise that there can be multiple θ that affect $p(x \mid \theta)$ and multiple ν that affect $p(\theta \mid \nu)$.

distribution with the point estimate, the posterior distribution is approximated so that it has mass only at the location of point estimate. The use of point estimates usually reduces the computational burden, but underestimating the uncertainty of θ. With the point estimate $\hat{\theta}$ for the parameter θ, the posterior predictive distribution of x^* is

$$\begin{aligned} p(x^* \mid x) &= \int p(x^* \mid \theta, x) p(\theta \mid x) d\theta, \\ &= p(x^* \mid \hat{\theta}, x). \end{aligned} \tag{2}$$

The idea of prior and posterior distributions also naturally extends to hierarchical models. Consider for example the posterior distribution representation in (1) where we have the model parameters θ. We can further extend this model specification by assuming that the prior distribution of θ is conditioned on some *hyperparameters* ν. Now by setting prior distribution for ν, *hyperprior* of the model, we can construct a hierarchical model where we have first layer parameters θ and second layer hyperparameters ν which affect the first layer parameters. The joint posterior distribution of the parameters and the hyperparameters is now

$$p(\theta, \nu \mid x) = \frac{p(x \mid \theta, \nu) p(\theta, \nu)}{p(x)} = \frac{p(x \mid \theta) p(\theta \mid \nu) p(\nu)}{\iint p(x \mid \theta) p(\theta \mid \nu) p(\nu) d\theta d\nu}, \tag{3}$$

where $p(\nu)$ is the prior distribution of ν. Figure 1 is the graphical representation of model in (3).

3 Models

3.1 Gaussian Processes

In this section, we go over the Gaussian process framework. Gaussian process (*e.g.* Rasmussen and Williams, 2006) is an infinite set of points for which any finite subset has a Gaussian joint distribution

$$p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{4}$$

The mean and covariance of the Gaussian distribution are defined by the mean and covariance functions of the Gaussian processes respectively:

$$\boldsymbol{\mu} = m(X), \tag{5}$$
$$\boldsymbol{\Sigma} = K(X, X), \tag{6}$$

where we have denoted the set of data points as $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$. Gaussian processes are mainly used as a prior distribution for some unknown functions by assuming that the process $f(\mathbf{x})$ is Gaussian. The prior mean function $m(X)$ is usually assumed to be zero due to notational convenience. The process $f(\mathbf{x})$, evaluated at the data points X, $f(X)$, can be compactly presented as

$$f(X) \sim \mathcal{GP}(0, K(X, X)), \tag{7}$$

where $K(X, X) = \mathbb{E}(f(X)^T f(X))$ denotes the covariance between function values. The assumption that $f(\mathbf{x})$ is a Gaussian process means that we assume an infinite dimensional normal distribution as a prior distribution for the function values. Furthermore, the covariance between function values depends only on the inputs at the corresponding locations. In practice, this prior assumption realizes as an n-dimensional normal distribution after we have n observations from the function. Let us set $\mathbf{f} = f(X)$ to be the vector containing the function values. Due to the nature of the Gaussian process, the joint distribution of function values $\mathbf{f}$, at data points X, and $\mathbf{f}^*$, at data points X_*, can be expressed as

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}^* \end{bmatrix} = \mathrm{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right). \tag{8}$$

From here on we denote $K_{**} = K(X_*, X_*), K_* = K(X_*)$ and $K = K(X, X)$. Now using properties of Gaussian distributions, the conditional distribution of $\mathbf{f}^*$ given X_*, X and $\mathbf{f}$ can be expressed as (see Appendix A)

$$\mathbf{f}^* \mid \mathbf{f}, X, X_* \sim \mathrm{N}(K_*^T K^{-1}\mathbf{f}, K_{**} - K_*^T K^{-1} K_*). \tag{9}$$

3.1.1 Mean and covariance functions

Gaussian process is defined by its mean and covariance functions.

In this thesis, we use zero prior mean function $m(X) = 0$ and the *Squared-Exponential* or *exponential quadratic* covariance function

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{\sum_{i=1}^{d}(x_i - x_i')^2}{2l_i}\right), \tag{10}$$

where σ_f^2 is signal variance or the *magnitude* , l_i are the characteristic *lengthscales* of the input-space and $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ denote single data points.

3.1.2 Regression

In regression tasks, the function $\mathbf{f}$ is not usually observed directly, but rather, we observe a noisy version of it

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\epsilon}, \tag{11}$$

where $\boldsymbol{\epsilon}$ is the noise term. It is often assumed that the noise is additive and independent zero mean Gaussian noise (white noise) with some variance σ^2, meaning that

$$\boldsymbol{\epsilon} \sim \mathrm{N}(0, \sigma^2 I), \tag{12}$$

leading to

$$\mathbf{y} \sim \mathrm{N}(\mathbf{f}, \sigma^2 I). \tag{13}$$

Now if we assume the Gaussian process prior for function values $\mathbf{f}$, so that

$$\mathbf{f} \sim \mathrm{N}(0, K), \tag{14}$$

the distribution of $\mathbf{y}$ can be expressed as (due to nature of Gaussian distributions)

$$\mathbf{y} \sim N(0, K + \sigma^2 I). \tag{15}$$

Recalling the joint distribution of $\mathbf{f}$ and $\mathbf{f}^*$ from the previous section, the joint distribution of $\mathbf{y}$ and $\mathbf{f}^*$ is

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} = \mathrm{N}\left(\mathbf{0}, \begin{bmatrix} K + \sigma^2 I & K_* \\ K_*^T & K_{**} \end{bmatrix}\right). \tag{16}$$

Using the properties of the Gaussian distributions, we can again condition the function values $\mathbf{f}^*$ on the observations $\mathbf{y}$ to get

$$\mathbf{f}^* \mid \mathbf{y}, X, X^* \sim \mathrm{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*,), \tag{17}$$

where

$$\boldsymbol{\mu}_* = K_*^T (K + \sigma^2 I)^{-1} \mathbf{y}, \tag{18}$$

$$\boldsymbol{\Sigma}_* = K_{**} - K_*^T (K + \sigma^2 I)^{-1} K_*. \tag{19}$$

Equations (17) – (19) can be used for predicting the function values $\mathbf{f}^*$ at locations X^* when we know the noisy function values $\mathbf{y}$ at locations X.

When we make the assumption that $\mathbf{y}$ are function values $\mathbf{f}$ with some Gaussian noise, we set the likelihood function of our model to be Gaussian

$$p(\mathbf{y} \mid \mathbf{f}) = \mathrm{N}(\mathbf{y} \mid \mathbf{f}, \sigma^2 I). \tag{20}$$

The conditional distribution $p(\mathbf{f}^* \mid \mathbf{y}, X, X^*)$ can be computed as

$$\begin{aligned} p(\mathbf{f}^* \mid \mathbf{y}, X, X^*) &= \int p(\mathbf{f}^*, \mathbf{f} \mid \mathbf{y}, X, X^*) d\mathbf{f}, & (21) \\ &= \int p(\mathbf{f}^* \mid \mathbf{f}, X, X^*) p(\mathbf{f} \mid \mathbf{y}, X) d\mathbf{f}, & (22) \\ &= \int p(\mathbf{f}^* \mid \mathbf{f}, X, X^*) \frac{p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid X)}{p(\mathbf{y} \mid X)} d\mathbf{f}, & (23) \end{aligned}$$

which can be computed analytically and results in (17). However, if our likelihood function is not Gaussian, the above integral cannot be evaluated analytically. In this case the only way to proceed is to either evaluate the integral numerically for example by using some sampling scheme, or approximate the appropriate components as Gaussian so the integral can be evaluated analytically.

The marginal likelihood, or *evidence*

$$Z = p(\mathbf{y} \mid X) = \int p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid X) d\mathbf{f}, \tag{24}$$

of the GP model is an important value in that it is often used for parameter estimation in GP framework. The parameters of the GP model are the parameters of the covariance functions and the parameters of the likelihood function. The parameters of the covariance function are usually called the *hyperparameters* of the GP model as they can be considered second layer parameter (the parameters of the prior distribution). The marginal likelihood can be used to find the optimal parameters of the GP model, because the maximum of the marginal likelihood usually corresponds with good predictions (Nickisch and Rasmussen, 2008; Riihimäki et al., 2013). The hyperparameters could also be selected by maximizing the predictive performance, but maximizing the marginal likelihood with respect to the hyperparameters usually results in better predictive performance globally. For a GP model with Gaussian likelihood, the log marginal likelihood can be computed analytically

$$\log p(\mathbf{y} \mid X) = -\frac{1}{2}\mathbf{y}^T (K + \sigma^2 I)^{-1} \mathbf{y} - \frac{1}{2} \log |K + \sigma^2 I| - \frac{n}{2} \log 2\pi, \tag{25}$$

by noting that $\mathbf{y} \sim \mathrm{N}(\mathbf{f}, \sigma^2) = \mathrm{N}(\mathbf{0}, K + \sigma^2 I)$.

3.1.3 Classification and other non-Gaussian likelihoods

Sometimes the observations are not function values $\mathbf{f}$ or even the noisy version directly, but rather something entirely different. Consider for example when the observations are whether a customer buys an item or not. In this case the observation is a binary variable, $y \in \{0, 1\}$. Another possibility could be number of customers in a bank during lunch hour. Now the observation y is a non-negative integer. For these example cases, the likelihood function $p(\mathbf{y} \mid \mathbf{f})$ is no longer Gaussian.

The statistician has to then decide an appropriate likelihood to model the observations. The number of the bank customers could be modelled as a Poisson distributed variable for example. Regardless of the case, the behaviour of the likelihood function is usually governed by some parameters. For the Poisson distribution, this is the expected number of occurrences or events. By transforming the appropriate parameter to the interval $(-\infty, \infty)$, Gaussian process prior can be used for the transformed variable, which is usually denoted as the *latent* function $\mathbf{f}$ (latent as in we don't directly observe it).

Assuming the GP prior for the function values $\mathbf{f}$ in these cases results in an analytically intractable posterior distribution $p(\mathbf{f} \mid X, \mathbf{y})$, as the likelihood and prior cannot be combined analytically. If we have an analytically intractable posterior distribution, we cannot evaluate analytically the predictive distribution of the function values

$$\begin{aligned} p(\mathbf{f}^* \mid \mathbf{y}, X, X^*) &= \int p(\mathbf{f}^* \mid \mathbf{f}, X, X^*) p(\mathbf{f} \mid X, \mathbf{y}) d\mathbf{f}, \\ &= \int p(\mathbf{f}^* \mid \mathbf{f}, X, X^*) \frac{p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid X)}{p(\mathbf{y} \mid X)} d\mathbf{f}, \end{aligned} \tag{26}$$

or the observations

$$p(\mathbf{y}^* \mid \mathbf{y}, X, X^*) = \int p(\mathbf{y}^* \mid \mathbf{f}^*) p(\mathbf{f}^* \mid X, X^*, \mathbf{y}) d\mathbf{f}^*. \tag{27}$$

To be able to do inference when the likelihood function is not Gaussian, we can either resort to sampling methods (Neal, 1997) to sample the posterior of the latent values $p(\mathbf{f} \mid \mathbf{y}, X)$ or approximate the posterior distribution as a Gaussian distribution

$$p(\mathbf{f} \mid \mathbf{y}, X) = \frac{1}{Z} N(\mathbf{f} \mid X) p(\mathbf{y} \mid \mathbf{f}) \approx q(\mathbf{f} \mid \mathbf{y}, X) = \mathrm{N}(\mathbf{f} \mid \boldsymbol{\mu}, \Sigma), \tag{28}$$

where $Z = \int N(\mathbf{f} \mid X) p(\mathbf{y} \mid \mathbf{f}) d\mathbf{f}$ is the marginal likelihood. In this thesis, we use two different approximation approaches, *expectation propagation* (Opper and Winther, 2000; Minka, 2001a,b) and variational inference to do the Gaussian approximation for the posterior distribution. In addition to the approximation methods, we also use sampling based methods for comparison.

3.1.4 Expectation Propagation

Often, we can assume that the data is independent and identically distributed (i.i.d.). This means that the joint likelihood $p(\mathbf{y} \mid \mathbf{f})$ factorizes over the observations

$$p(\mathbf{y} \mid \mathbf{f}) = \prod_{i=1}^{n} p(y_i \mid f_i), \tag{29}$$

where n is the number of observations. Expectation propagation (EP, Opper and Winther, 2000; Minka, 2001a,b) approximates the posterior distribution of the latent function values $\mathbf{f}$ as a Gaussian distribution by replacing the non-Gaussian likelihood terms $p(y_i \mid f_i)$ in (29) with unnormalized Gaussian *site approximations* $\tilde{Z}_i \tilde{t}_i(f_i)$. The joint likelihood can now be expressed as

$$p(\mathbf{y} \mid \mathbf{f}) \approx \prod \tilde{Z}_i \tilde{t}_i(f_i) = \prod \tilde{Z}_i \mathrm{N}(f_i \mid \tilde{\mu}_i, \tilde{\sigma}_i^2) = \tilde{Z}\mathrm{N}(\mathbf{f} \mid \tilde{\boldsymbol{\mu}}, \tilde{\Sigma}), \tag{30}$$

where $\tilde{\Sigma} = \mathrm{diag}(\tilde{\sigma}_1^2, \tilde{\sigma}_2^2, \ldots, \tilde{\sigma}_n^2)$ and $\tilde{\boldsymbol{\mu}} = [\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_n]^T$. Combining the GP prior of $\mathbf{f}$ and the site approximations results in a Gaussian posterior

$$p(\mathbf{f} \mid \mathbf{y}, X) \approx \frac{1}{Z_{ep}} p(\mathbf{f} \mid X) \prod_{i=1}^{n} \tilde{t}_i(f_i) = q(\mathbf{f} \mid \mathbf{y}, X), \tag{31}$$

where $q(\mathbf{f} \mid \mathbf{y}, X) = \mathrm{N}(\mathbf{f} \mid \boldsymbol{\mu}, \Sigma)$ and Z_{ep} is the EP approximation to the marginal likelihood. The parameters of the posterior approximation can be expressed as

$$\boldsymbol{\mu} = \Sigma \tilde{\Sigma} \tilde{\boldsymbol{\mu}}, \tag{32}$$

$$\Sigma = \left(K^{-1} + \tilde{\Sigma}^{-1} \right)^{-1}. \tag{33}$$

After initializing the site approximations to some appropriate values, the following process is iterated through all i until convergence to achieve the EP approximation to the posterior distribution:

(i) Form the cavity distribution $q_{-i}(f_i)$ by removing the ith site approximation $\tilde{t}_i$ from the marginal posterior distribution $q_i(f_i)$

$$q_{-i}(f_i) = \mathrm{N}(f_i \mid \mu_{-i}, \sigma_{-i}^2) \propto \frac{q_i(f_i)}{\tilde{t}_i(f_i)}, \tag{34}$$

with

$$\mu_{-i} = \sigma_{-i}^2 \left(\frac{\mu_i}{\sigma_i^2} - \frac{\tilde{\mu}_i}{\tilde{\sigma}_i^2} \right),$$

$$\sigma_{-i}^2 = \left(\frac{1}{\sigma_i^2} - \frac{1}{\tilde{\sigma}_i^2} \right).$$

(ii) Find the new marginal posterior approximation for f_i by minimizing the *Kullback-Leibler* (KL) divergence (Kullback and Leibler, 1951) from the marginal posterior approximation to the *tilted* distribution $\hat{p}(f_i) = q_{-i}(f_i) p(y_i \mid f_i)$.

$$\min_q \mathrm{KL}(q_i(f_i) \parallel \hat{p}(f_i)). \tag{35}$$

For Gaussian approximation $q_i(f_i)$, this means that we set the first two moments of the new marginal posterior approximation to be the first two moments of the tilted distribution

$$\mathbb{E}_q[f_i] = \int f_i q_{-i}(f_i) p(y_i \mid f_i) df_i,$$

$$\mathbb{E}_q[f_i^2] = \int f_i^2 q_{-i}(f_i) p(y_i \mid f_i) df_i.$$

(iii) Compute the normalization term $\hat{Z}_i$ of $\hat{p}(f_i)$

$$\hat{Z}_i = \int q_{-i}(f_i) p(y_i \mid f_i) df_i.$$

(iv) Update the site approximation by removing the cavity distribution $q_{-i}(f_i)$ from the new marginal posterior approximation

$$\tilde{t}_i(f_i) = \mathrm{N}(f_i \mid \tilde{\mu}_i, \tilde{\sigma}_i^2) \propto \frac{q_i(f_i)}{q_{-i}(f_i)},$$

with

$$\begin{aligned} \tilde{\mu}_i &= \tilde{\sigma}_i^2 \left(\frac{\mu_i}{\sigma_i^2} - \frac{\mu_{-i}}{\sigma_{-i}^2} \right), \\ \tilde{\sigma}_i^2 &= \left(\frac{1}{\sigma_i^2} - \frac{1}{\sigma_{-i}^2} \right). \end{aligned}$$

(v) Compute the normalization terms $\tilde{Z}_i$. This can be done by noting that the moments of $\tilde{t}_i(f_i) q_{-i}(f_i)$ must match the moments of $\hat{p}(f_i)$ or $q_i(f_i)$

$$\begin{aligned} \tilde{Z}_i &= \hat{Z}_i \int \mathrm{N}(f_i \mid \tilde{\mu}_i, \tilde{\sigma}_i^2) \mathrm{N}(f_i \mid \mu_{-i}, \sigma_{-i}^2) df_i, && (36) \\ &= \hat{Z}_i (2\pi)^{1/2} (\tilde{\sigma}_i^2 + \sigma_{-i}^2)^{1/2} \exp\left(\frac{(\tilde{\mu}_i - \mu_{-i})^2}{2(\tilde{\sigma}_i^2 + \sigma_{-i}^2)} \right), && (37) \end{aligned}$$

where we have used (B.4) to compute the normalization constant for the product of the two Gaussian distributions.

Marginal likelihood with EP can be computed analogously to the regression case

$$p(\mathbf{y} \mid X) = \int p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid X) d\mathbf{f} \approx \left(\prod_{i=1}^{n} \tilde{Z}_i \right) \int p(\mathbf{f} \mid X) \prod_{i=1}^{n} \tilde{t}_i(f_i). \tag{38}$$

The part inside the integral corresponds to the marginal likelihood in the regression case. The log marginal likelihood can be expressed as

$$\begin{aligned} \log Z = \sum_{i=1}^{n} \log \hat{Z}_i + \frac{1}{2} \log(\tilde{\sigma}_i^2 + \sigma_{-i}^2) + \frac{(\tilde{\mu}_i - \mu_{-i})^2}{2(\tilde{\sigma}_i^2 + \sigma_{-i}^2)} && (39) \\ - \frac{1}{2} \log |K + \tilde{\Sigma}| - \frac{1}{2} \tilde{\boldsymbol{\mu}}^T (K + \tilde{\Sigma})^{-1} \tilde{\boldsymbol{\mu}}. && (40) \end{aligned}$$

3.2 Gaussian Processes with monotonicity constraint

Sill and Abu-Mostafa (1997) proposed a general framework for including monotonicity constraints for multilayer perceptron (see for example Bishop 2006) by using appropriately placed virtual observations or *hints* to force monotonicity. Standard Gaussian process prior assumption on $\mathbf{f}$ does not restrict the function values to be monotonically increasing or decreasing with respect to input dimensions. However, it is possible to use virtual observations with GP prior to force the monotonicity of the posterior of the latent values $p(\mathbf{f} \mid X, \mathbf{y})$ (Riihimäki and Vehtari, 2010).

Due to the nature of the Gaussian process and differentiation operation, the derivative of the GP (with respect to one of the input dimensions) is also a Gaussian process (*e.g.* Rasmussen, 2003; Solak et al., 2003). For example, the derivative of the expected value of function value f_i is

$$\frac{\partial \mathbb{E}[f_i]}{\partial x_i^{(d)}} = \mathbb{E}\left[\frac{\partial f_i}{\partial x_i^{(d)}}\right]. \tag{41}$$

Because the derivative of the Gaussian process is a Gaussian process, we can include virtual observations from the derivative to our GP model. As the differentiation is a linear operation, the covariance between derivative and the function value can be computed with

$$\mathrm{Cov}\left[\frac{\partial f_j}{\partial x_i^{(d)}}, f_i\right] = \frac{\partial}{\partial x_i^{(i)}} \mathrm{Cov}\left[f_j, f_i\right], \tag{42}$$

and covariance between derivatives

$$\mathrm{Cov}\left[\frac{\partial f_j}{\partial x_j^{(d)}}, \frac{\partial f_i}{\partial x_i^{(g)}}\right] = \frac{\partial^2}{\partial x_j^{(d)} \partial x_i^{(g)}} \mathrm{Cov}\left[f_j, f_i\right]. \tag{43}$$

The above derivatives are in Appendix A for Squared-Exponential covariance functions. The expected value or prediction of the derivative of the function can be expressed as

$$\mathbb{E}\left[\frac{\partial f^*}{\partial x_d^*}\right] = \frac{\partial K(\mathbf{x}^*, X)}{\partial x_d^*} \left(K(X,X) + \sigma^2\right)^{-1} \mathbf{y}. \tag{44}$$

As mentioned, the monotonicity is forced by adding virtual observations around the input space, and forcing the function to be either increasing or decreasing in those virtual inputs, with respect to one or more input dimensions. What this means is that we have virtual observations $\{\mathbf{x}_v, y_v\}$ where the response variable is $y_v = \{-1, 1\}$, depending on whether the function is decreasing or increasing with respect to one of the input dimensions. The monotonicity can now be integrated into the model by choosing a suitable likelihood function for the virtual observations. In this thesis, we use the *probit* likelihood

$$p\left(y_v \mid \frac{\partial f_v}{\partial x_v^d}\right) = \Phi(y_v \frac{\partial f_v}{\partial x_v^d}) = \int_{-\infty}^{\frac{\partial f_v}{\partial x_v^d}} N(t \mid 0, 1) dt. \tag{45}$$

The full posterior of the function values is

$$p(\mathbf{f}, \mathbf{f}' \mid X, X_v, \mathbf{y}, \mathbf{y}_v) = \frac{1}{Z} p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{y}_v \mid \mathbf{f}') p(\mathbf{f}, \mathbf{f}' \mid X, X_v), \tag{46}$$

where $\mathbf{f}'$ is the derivative of $\mathbf{f}$, Z is the marginal likelihood and $p(\mathbf{f}, \mathbf{f}' \mid X, X_v)$ is the GP prior of the function and its derivative

$$p(\mathbf{f}, \mathbf{f}' \mid X, X_v) = N\left(\mathbf{f}_{\text{joint}} \mid \mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X_v) \\ K(X_v,X) & K(X_v,X_v) \end{bmatrix}\right), \tag{47}$$

and

$$\mathbf{f}_{\text{joint}} = \begin{bmatrix} \mathbf{f} \\ \mathbf{f}' \end{bmatrix} \in \mathbb{R}^{n+m}. \tag{48}$$

If we have multidimensional input space, the monotonicity of the specific dimensions is incorporated into the prior covariance of $\mathbf{f}_{\text{joint}}$. The posterior is not analytically tractable when the likelihood is not Gaussian (see section 3.1.4). However, we can still utilize the methods in section 3.1.4 for approximating the posterior $p(\mathbf{f}'_{joint} \mid X, X_v, \mathbf{y}, \mathbf{y}_v)$ with a Gaussian distribution. This is done by approximating the non-Gaussian likelihood terms $p(y_{j,v} \mid f'_j)$, where $j = \{1, 2, \ldots, m\}$, with Gaussian site terms $\tilde{t}_j(f'_j)$. The Gaussian approximation to the posterior is now

$$\begin{aligned} q(\mathbf{f}_{\text{joint}} \mid X, X_v, \mathbf{y}, \mathbf{y}_v) &= \frac{1}{Z_{\text{EP}}} p(\mathbf{f}_{\text{joint}} \mid X, X_v) \prod_{i=1}^{n} p(y_i \mid f_i) \prod_{j=1}^{m} \tilde{Z}_j \tilde{t}_i(f'_j), \\ &= \mathrm{N}\left(\mathbf{f}_{\text{joint}} \mid \boldsymbol{\mu}_{\text{joint}}, \Sigma_{\text{joint}}\right), \end{aligned} \tag{49}$$

where $\boldsymbol{\mu}_{\text{joint}} = \Sigma_{\text{joint}} \tilde{\Sigma}_{\text{joint}}^{-1} \tilde{\boldsymbol{\mu}}_{\text{joint}}$, $\Sigma_{\text{joint}} = \left(K_{\text{joint}}^{-1} + \tilde{\Sigma}_{\text{joint}}^{-1}\right)^{-1}$ and

$$\tilde{\boldsymbol{\mu}}_{\text{joint}} = \begin{bmatrix} \mathbf{y} \\ \tilde{\boldsymbol{\mu}} \end{bmatrix}, \tag{50}$$

$$\tilde{\Sigma}_{\text{joint}} = \begin{bmatrix} \sigma^2 I & \mathbf{0} \\ \mathbf{0} & \tilde{\Sigma} \end{bmatrix}, \tag{51}$$

when $\tilde{t}_j(f'_j) = \mathrm{N}(f'_j \mid \tilde{\mu}_j, \tilde{\sigma}_j^2)$ and $\tilde{\boldsymbol{\mu}} = [\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_m]^T$, $\tilde{\Sigma} = \mathrm{diag}(\tilde{\sigma}_1^2, \tilde{\sigma}_2^2, \ldots, \tilde{\sigma}_m^2)$ as in section 3.1.4. The EP approximation to the marginal likelihood in the monotonic GP case is

$$p(\mathbf{y}, \mathbf{y}_v \mid X, X_v) = \left(\prod_{j=1}^{m} \tilde{Z}_j\right) \int p(\mathbf{f}, \mathbf{f}' \mid X, X_v) \prod_{i=1}^{n} p(y_i \mid f_i) \prod_{j=1}^{m} \tilde{t}_j(f'_j) d\mathbf{f}_{\text{joint}}, \tag{52}$$

and the log marginal likelihood is

$$\begin{aligned} \log Z_{\text{EP}} = \sum_{i=1}^{m} \left(\log \hat{Z}_i + \frac{1}{2} \log(\tilde{\sigma}_i^2 + \sigma_{-i}^2) + \frac{(\tilde{\mu}_i - \mu_{-i})^2}{2(\tilde{\sigma}_i^2 + \sigma_{-i}^2)} \right) + \frac{m-n}{2} \log 2\pi \\ - \frac{1}{2} \log |K_{\text{joint}} + \tilde{\Sigma}_{\text{joint}}| - \frac{1}{2} \tilde{\boldsymbol{\mu}}_{\text{joint}}^T (K_{\text{joint}} + \tilde{\Sigma}_{\text{joint}})^{-1} \tilde{\boldsymbol{\mu}}_{\text{joint}}. \end{aligned} \tag{53}$$

4 Methods

In this section we introduce the rest of the methods of this thesis and how these can be combined with models and methods of the previous section to enable monotonic Gaussian processes with large data sets. First we introduce the sparse Gaussian processes which enable the use of larger data sets with GP models. After this we review the variational inference framework, how this can be applied to larger data sets with stochastic variational inference, and finally the application of stochastic variational inference for Gaussian processes.

4.1 Sparse Gaussian Processes

The standard Gaussian process approach offers an intuivite, flexible and analytical framework for full probabilistic analysis. However, when using the approach introduced in section 3.1, the computational cost of doing the inference scales cubically with respect to the number of observations. The cubic scaling comes from inverting and computing the Cholesky factor of the covariance matrix $K(X,X) \in \mathbb{R}^{n \times n}$. This $O(n^3)$ computational cost can easily become too expensive when the number of observations increases. With modern desktop computers, the $O(n^3)$ cost prohibits the use of standard GP models for more than ten thousand observations.

To be able to use GP models with more than a few thousand data points, several sparse approximations have been proposed in the literature (see *e.g.* Quinonero-Candela and Rasmussen, 2005). All of these sparse methods have one thing in common: to do the inference exactly for m variables and to do approximate inference for the rest of the n variables ($m << n$). In practice this is done with the help of *inducing variables* $\mathbf{u} = [u_1, u_2, \dots, u_m]$. The inducing variables are the function values corresponding to *inducing inputs* X_u and are located in the same space as the function values $\mathbf{f}$ or $\mathbf{f}^*$. Due to the properties of the Gaussian processes, the prior for $\mathbf{f}$ or $\mathbf{f}^*$ can be expressed by marginalizing the inducing variables

$$\begin{aligned} p(\mathbf{f}, \mathbf{f}^* \mid X, X_*) &= \int p(\mathbf{f}, f^*, \mathbf{u} \mid X, X_*, X_u) d\mathbf{u}, && (54) \\ &= \int p(\mathbf{f} \mid \mathbf{u}, X) p(\mathbf{f}^* \mid \mathbf{u}, X_*) p(\mathbf{u} \mid X_u) d\mathbf{u}, && (55) \end{aligned}$$

where $p(\mathbf{u} \mid X_u)$ is just the GP prior of $\mathbf{u}$

$$p(\mathbf{u} \mid X_u) = \mathrm{N}\left(\mathbf{u} \mid \mathbf{0}, K(X_u, X_u)\right) = \mathrm{N}(\mathbf{u} \mid, \mathbf{0}, K_{uu}). \quad (56)$$

The conditional distributions $p(\mathbf{f} \mid \mathbf{u}, X)$ and $p(\mathbf{f}^* \mid \mathbf{u}, X_*)$ are

$$\begin{aligned} p(\mathbf{f} \mid \mathbf{u}, X) &= \mathrm{N}(\mathbf{f} \mid K_{nu} K_{uu}^{-1} \mathbf{u}, K - K_{nu} K_{uu}^{-1} K_{un}), && (57) \\ p(\mathbf{f}^* \mid \mathbf{u}, X_*) &= \mathrm{N}(\mathbf{f}^* \mid K_{*u} K_{uu}^{-1} \mathbf{u}, K_{**} - K_{*u} K_{uu}^{-1} K_{u*}), && (58) \end{aligned}$$

where we have used $K_{nu} = K(X, X_u) = K_{un}^T$ and $K_{*u} = K(X_*, X_u) = K_{u*}^T$. Most of the sparse methods proposed in the literature use exact prior $p(\mathbf{u} \mid X_u)$ and approximate the conditional distributions $p(\mathbf{f} \mid \mathbf{u}, X)$ and $p(\mathbf{f}^* \mid \mathbf{u}, X_*)$. We focus on the *Fully independent conditional* (FIC), also known as Sparse Gaussian processes using Pseudo-Inputs (SGPP, Snelson and Ghahramani, 2006), as it gives intuition on how the sparse approximations work and helps understand the variational approach later on.

4.1.1 Fully independent conditional

With FIC, the conditional distributions (57) and (58) are approximated with

$$q_{\text{FIC}}(\mathbf{f} \mid \mathbf{u}) = \prod_{i=1}^{n} p(f_i \mid \mathbf{u}) = \mathrm{N}\left(K_{nu}K_{uu}^{-1}\mathbf{u}, \operatorname{diag}(K - K_{nu}K_{uu}^{-1}K_{un})\right), \tag{59}$$

$$q_{\text{FIC}}(\mathbf{f}^* \mid \mathbf{u}) = \prod_{i=1}^{n_*} p(f_i^* \mid \mathbf{u}) = \mathrm{N}\left(K_{*u}K_{uu}^{-1}\mathbf{u}, \operatorname{diag}(K_{**} - K_{*u}K_{uu}^{-1}K_{u*})\right), \tag{60}$$

so that we use the exact mean for the conditional distribution but approximate the covariance matrix with the exact diagonal (i.e. independent terms). The approximative priors $q_{\text{FIC}}(\mathbf{f} \mid X)$ and $q_{\text{FIC}}(\mathbf{f}^* \mid X_*)$ can now be given as

$$q_{\text{FIC}}(\mathbf{f} \mid X) = \int q_{\text{FIC}}(\mathbf{f} \mid \mathbf{u})p(\mathbf{u} \mid X_u)d\mathbf{u}, \tag{61}$$

$$= \mathrm{N}(\mathbf{f} \mid \mathbf{0}, K_{nu}K_{uu}^{-1}K_{un} - \operatorname{diag}(K - K_{nu}K_{uu}^{-1}K_{un})), \tag{62}$$

$$q_{\text{FIC}}(\mathbf{f}^* \mid X_*) = \int q_{\text{FIC}}(\mathbf{f}^* \mid \mathbf{u})p(\mathbf{u} \mid X_u)d\mathbf{u}, \tag{63}$$

$$= \mathrm{N}(\mathbf{f}^* \mid \mathbf{0}, K_{*u}K_{uu}^{-1}K_{u*} - \operatorname{diag}(K - K_{*u}K_{uu}^{-1}K_{u*})). \tag{64}$$

4.2 Variational Inference

Variational inference or *Variational Bayes* is a method, like EP, to approximate the true posterior distribution $p(\mathbf{f} \mid X, \mathbf{y})$ and marginal likelihood $p(\mathbf{y} \mid X)$ (Bishop, 2006). Assuming $q(\mathbf{f} \mid X, \mathbf{y})$ is the approximation to the true posterior distribution $p(\mathbf{f} \mid X, \mathbf{y})$. The log marginal likelihood $p(\mathbf{y} \mid X)$ can be decomposed to

$$\log p(\mathbf{y} \mid X) = \mathcal{L}(q) + \mathrm{KL}(q \parallel p), \tag{65}$$

where

$$\mathcal{L}(q) = \int q(\mathbf{f} \mid X, \mathbf{y}) \log \frac{p(\mathbf{f}, \mathbf{y} \mid X)}{q(\mathbf{f} \mid X, \mathbf{y})} d\mathbf{f}, \tag{66}$$

$$\mathrm{KL}(q \parallel p) = -\int q(\mathbf{f} \mid X, \mathbf{y}) \log \frac{p(\mathbf{f} \mid X, \mathbf{y})}{q(\mathbf{f} \mid X, \mathbf{y})} d\mathbf{f}. \tag{67}$$

The decomposition (65) can be realized by first noting that

$$\log p(\mathbf{f}, \mathbf{y} \mid X) = \log p(\mathbf{f} \mid X, \mathbf{y}) + \log p(\mathbf{y} \mid X). \tag{68}$$

Now by plugging in (68) to the definition of $\mathcal{L}(q)$ in (66), we get

$$\begin{aligned}
\mathcal{L}(q) &= \int q(\mathbf{f} \mid X, \mathbf{y}) \left(\log p(\mathbf{f} \mid X, \mathbf{y}) + \log p(\mathbf{y} \mid X) - \log q(\mathbf{f} \mid X, \mathbf{y})\right) d\mathbf{f}, && (69)\\
&= \int q(\mathbf{f} \mid X, \mathbf{y}) \log \frac{p(\mathbf{f} \mid X, \mathbf{y})}{q(\mathbf{f} \mid X, \mathbf{y})} d\mathbf{f} + \int \log p(\mathbf{y} \mid X) q(\mathbf{f} \mid X, \mathbf{y}) d\mathbf{f}, && (70)\\
&= -\mathrm{KL}(q \parallel p) + \log p(\mathbf{y} \mid X). && (71)
\end{aligned}$$

If we want to maximize the marginal likelihood $p(\mathbf{y} \mid X)$ with respect to $q(\mathbf{f} \mid X, \mathbf{y})$, it is enough to maximize $\mathcal{L}(q)$ or equivalently, minimize $\mathrm{KL}(q \parallel p)$. If we don't restrict $q(\mathbf{f} \mid X, \mathbf{y})$, it is evident that the KL-divergence minimizes when $q(\mathbf{f} \mid X, \mathbf{y}) = p(\mathbf{f} \mid X, \mathbf{y})$. However, because this does not offer anything in the approximative sense (tractability), we restrict $q(\mathbf{f} \mid X, \mathbf{y})$ to be something simpler that has tractable form and try to minimize the KL-divergence.

Altough there is an infinite number of possible ways to restrict the distribution $q(\mathbf{f} \mid X, y)$, one often used is the factorized approximation

$$q(\mathbf{f} \mid X, \mathbf{y}) = \prod_{i=1}^{n} q_i(\mathbf{f}_i \mid X, \mathbf{y}), \tag{72}$$

which corresponds to the *mean field theory* in physics (Parisi, 1998). The idea here is that the independent factors q_i are not restricted and we now seek the q_i, from all possible distributions, that maximize the $\mathcal{L}(q)$. It can be shown that the q_i which maximize $\mathcal{L}(q)$ are (Bishop, 2006, pp. 464-466)

$$\log q_i(f_i \mid X, \mathbf{y}) = \mathbb{E}_{q_j, j \neq i}[\log p(\mathbf{f}, \mathbf{y} \mid X)] + \text{constant}. \tag{73}$$

Note that as the $\log q_i(f_i \mid X, \mathbf{y})$ depends on other terms q_j, some sort of cycling procedure might be needed where each term is updated in turn until some convergence criterion is satisfied. These factorized approximations have the nice property that the convergence is guaranteed, because the bound is convex with respect to all of the factors q_i (Boyd and Vandenberghe, 2004).

4.3 Variational Inference with mean field approximation

Now consider a model where we have two sets of latent variables: Vector of global latent variables $\mathbf{u}$ and local latent variables $\mathbf{z} = [z_1, z_2, \ldots, n]$. Each of which is a vector of J elements, $z_i = [z_{i.1}, z_{i,2}, \ldots, z_{i,J}]$. In addition to the latent variables, we have n observations $\mathbf{y} = [y_1, y_2, \ldots, y_n]$. We suppress the conditioning to X for the distributions to clear the notation. We assume that the observations y_i and local latent variables z_i are independent of the other observations $\mathbf{y}_{-i} = [y_1, y_2, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n]$ and local latent variables $\mathbf{z}_{-i} = [z_1, z_2, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n]$, given the global latent variables $\mathbf{u}$:

$$p(y_i, z_i \mid \mathbf{y}_{-i}, \mathbf{z}_{-i}, \mathbf{u}) = p(y_i, z_i \mid \mathbf{u}). \tag{74}$$

In addition to the independence, we assume that the conditional distributions of the latent variables are in the exponential family:

$$p(\mathbf{u} \mid \mathbf{y}, \mathbf{z}) = h(\mathbf{u}) \exp\left(\eta_g(\mathbf{y}, \mathbf{z})^T t(\mathbf{u}) - a_g(\eta_g(\mathbf{y}, \mathbf{z}))\right), \tag{75}$$

$$p(z_{i,j} \mid y_i, z_{i,-j}, \mathbf{u}) = h(z_{i,j}) \exp\left(\eta_l(y_i, z_{i,-j}, \mathbf{u})^T t(z_{i,j}) - a_l(\eta_l(y_i, z_{i,-j}, \mathbf{u}))\right), \tag{76}$$

where functions $h(\cdot)$ and $a(\cdot)$ are *base measure* and *log-normalizer* and the vector functions $\eta(\cdot)$ and $t(\cdot)$ are the *natural parameter* and *sufficient statistics*. Given the assumptions, the distribution of the local variables, conditioned on the global latent variables, must also be of the exponential family, specifically:

$$p(y_i, z_i \mid \mathbf{u}) = h(y_i, z_i) \exp\left(\mathbf{u}^T t(y_i, z_i) - a_l(\mathbf{u})\right), \tag{77}$$

as do the prior of the global latent variables

$$p(\mathbf{u}) = h(\mathbf{u}) \exp\left(\boldsymbol{\alpha}^T t(\mathbf{u}) - a_g(\boldsymbol{\alpha})\right). \tag{78}$$

In section 4.2 the log marginal likelihood was expressed in the decomposed form. We can extend and apply the same procedure now for two sets of latent variables $\mathbf{z}$ and $\mathbf{u}$

$$\log p(\mathbf{y}) = L(q) + \mathrm{KL}(q \parallel p) \tag{79}$$

$$\geq L(q) = \int q(\mathbf{z}, \mathbf{u}) \log \frac{p(\mathbf{y}, \mathbf{z}, \mathbf{u})}{q(\mathbf{z}, \mathbf{u})} d\mathbf{z} d\mathbf{u}, \tag{80}$$

because KL divergence is always positive. As noted earlier, the idea in variational inference is to restrict the variational distribution $q(\mathbf{z}, \mathbf{u})$ so that we can compute the expectations efficiently in the maximization of the lower bound $L(q)$.

To be able to do the inference with large data sets, we assume that $q(\mathbf{z}, \mathbf{u})$ is of the mean field family, meaning that it factorizes

$$q(\mathbf{z}, \mathbf{u}) = q(\mathbf{u} \mid \theta) \prod_{i=1}^{n} \prod_{j=1}^{J} q(z_{i,j} \mid \phi_{i,j}), \tag{81}$$

where we have noted the parameters of the global latent variables as θ and the parameters of the local latent varibles with $\phi_{i,j}$. For tractability, we set the factors $q(\mathbf{u} \mid \theta)$ and $q(z_{i,j} \mid \phi_{i,j})$ to be the same exponential family distributions as the conditional distributions $p(\mathbf{u} \mid \mathbf{y}, \mathbf{z})$ and $p(z_{i,j} \mid y_i, z_{i,-j}, \mathbf{u})$.

$$q(\mathbf{u} \mid \theta) = h(\mathbf{u}) \exp\left(\theta^T t(\mathbf{u}) - a_g(\theta)\right) \tag{82}$$

$$q(z_{i,j} \mid \phi_{i,j}) = h(z_{i,j}) \exp\left(\phi_{i,j}^T t(z_{i,j}) - a_l(\phi_{i,j})\right) \tag{83}$$

With the help of mean field approximation (81), the marginal likelihood lower bound $L(q)$ can now be written as a function of θ

$$L(\theta) = \int q(\mathbf{z}, \mathbf{u}) \log \frac{p(\mathbf{u} \mid \mathbf{y}, \mathbf{z}) p(\mathbf{y}, \mathbf{z})}{q(\mathbf{u} \mid \theta) q(\mathbf{z})} d\mathbf{z} d\mathbf{u}, \tag{84}$$

$$= \int \log p(\mathbf{u} \mid \mathbf{y}, \mathbf{z}) q(\mathbf{z}, \mathbf{u}) d\mathbf{u} d\mathbf{z} - \int \log q(\mathbf{u} \mid \theta) q(\mathbf{z}, \mathbf{u}) d\mathbf{u} d\mathbf{z} + \text{constant}, \tag{85}$$

where the constant term absorbs all the terms that do not depend on $\mathbf{u}$. Substituting the conditional distributions $p(\mathbf{u} \mid \mathbf{y}, \mathbf{z})$ and $q(\mathbf{u} \mid \theta)$ from (75) and (76) to the lower bound (85), we get

$$L(\theta) = \left(\int \eta_g(\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha}) q(\mathbf{z}, \mathbf{u}) d\mathbf{u} d\mathbf{z} \right)^T \nabla_\theta a_g(\theta) - \theta^T \nabla_\theta a_g(\theta) + a_g(\theta) + \text{constant}, \quad (86)$$

where we have used the fact that the expectation of the sufficient statistics is the gradient of the log-normalizer, $\int t(\mathbf{u}) q(\mathbf{z}, \mathbf{u}) = \nabla_\theta a_g(\theta)$. The bound can now be maximized with respect to the global variational parameters θ by taking the gradient of (86) with respect to θ

$$\nabla_\theta L(\theta) = \nabla_\theta^2 a_g(\theta) \left(\mathbb{E}_q[\eta_g(\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha})] - \theta \right), \quad (87)$$

and setting it to zero, to get

$$\theta = \mathbb{E}[\eta_g(\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha})]. \quad (88)$$

With the same kind of procedure, the gradient of the bound with respect to local variational parameters $\phi_{i,j}$ is

$$\nabla_{\phi_{i,j}} L(\phi_{i,j}) = \nabla_{\phi_{i,j}}^2 a_l(\phi_{i,j}) \left(\mathbb{E}_q[\eta_l(y_i, z_{i,-j}, \mathbf{u}) - \phi_{i,j} \right), \quad (89)$$

which equal zero when

$$\phi_{i,j} = \mathbb{E}[\eta_l(y_i, z_{i,-j}, \mathbf{u})]. \quad (90)$$

The above equations form the algorithm for coordinate ascent variational inference with mean-field approximation, iterating between the updates of the local and global parameters. The problem with the standard coordinate ascent method is that the updates of the global parameters need the whole data to be available.

4.4 Stochastic Variational Inference

When the data sets get increase in size, problems arise with the standard variational approach. Consider for example the case where we have hundreds of thousands of observations. Computing the expectations can become very expensive for all but the simplest models. Hoffman et al. (2013) proposed *stochastic variational inference* (SVI) framework which enables application of variational inference for large data sets.

Stochastic variational inference uses stochastic *natural gradients* (Amari, 1998) computed using only the subset of the data for updating the global variational parameters. Natural gradients work in the space of KL divergence between two distributions, rather than the euclidean space of the parameters as the normal gradients do. The natural gradient of the objective function can be computed by premultiplying the gradient with inverse of the Fisher information matrix (Amari, 1982; Kullback and Leibler, 1951)

$$\hat{\nabla}_\theta f(\theta) = G(\theta)^{-1} \nabla_\theta f(\theta), \quad (91)$$

where G is the Fisher information matrix of $q(\theta)$

$$G(\theta) = \mathbb{E}_\theta[\nabla_\theta \log q(\mathbf{u} \mid \theta)(\log q(\mathbf{u} \mid \theta))^T]. \tag{92}$$

When $q(\mathbf{u} \mid \theta)$ is in the exponential family, the Fisher information matrix is the second derivative of the log normalizer

$$\begin{aligned} G(\theta) &= \mathbb{E}_\theta[(\nabla_\theta \log p(\mathbf{u} \mid \theta))(\nabla_\theta \log p(\mathbf{u} \mid \theta))^T], && (93)\\ &= \mathbb{E}_\theta[(t(\mathbf{u}) - \mathbb{E}[t(\mathbf{u})])(t(\mathbf{u}) - \mathbb{E}_\theta[t(\mathbf{u})])^T], && (94)\\ &= \nabla^2_\theta a_g(\theta). && (95)\end{aligned}$$

We can now compute the natural gradients of the lower bound $L(\theta)$ and $L(\phi_{i,j})$ from the gradients in equations (87) and (89)

$$\begin{aligned} \hat{\nabla}_\theta L(\theta) &= (\nabla^2_\theta a_g(\theta))^{-1} \nabla_\theta L(\theta) = \mathbb{E}_q[\eta_g(\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha})] - \theta, && (96)\\ \hat{\nabla}_\theta L(\phi_{i,j}) &= (\nabla^2_{\phi_{i,j}} a_l(\phi_{i,j}))^{-1} \nabla_{\phi_{i,j}} L(\phi_{i,j}) = \mathbb{E}_q[\eta_l(y_i, z_{i,-j}, \mathbf{u}) - \phi_{i,j}. && (97)\end{aligned}$$

We see that taking a step of unit length towards the direction of natural gradient corresponds to coordinate ascent update of the parameters.

Stochastic variational inference works by replicating the subsets of the data to form noisy estimates for the natural gradients. This can be effective when we can write the objective function as a sum of terms, as in the variational inference. The stochastic gradient methods have been proven to converge to an optimum (Robbins and Monro, 1951) given an appropriate step size schedule. For convex objective functions, this optimum is global, and for non-convex functions, local or global. Consider objective function $L(\theta)$ and its noisy gradient $\nabla_\theta L(\theta)$, computed with the replicated subset of the data. In standard coordinate ascent, or *gradient ascent*, we take a step of length $\rho^{(i)}$ towards the direction of the gradient to update the parameter estimate

$$\theta^{(i+1)} = \theta^{(i)} + \rho^{(i)} \nabla_\theta L(\theta^{(i)}). \tag{98}$$

If the gradient is the noisy estimate of the true gradient, the *stochastic gradient ascent* is guaranteed to converge to local optimum of the objective function if the step size $\rho^{(i)}$ satisfies the following

$$\sum \rho^{(i)} = \infty, \qquad \sum \left(\rho^{(i)}\right)^2 < \infty. \tag{99}$$

The above condition also applies for the natural gradients

$$\theta^{(i+1)} = \theta^{(i)} + \rho^{(i)} G_i^{-1} \nabla_\theta L(\theta^{(i)}) = \theta^{(i)} + \rho^{(i)} \hat{\nabla}_\theta L(\theta^{(i)}). \tag{100}$$

Consider sampling the ith data point from the whole dataset. The stochastic variational inference updates the global variational parameters θ with

$$\theta(i+1) = \theta^{(i)} + \rho^{(i)} \hat{\nabla}_\theta L(\theta^{(i)}) = \theta^{(i)} + \rho^{(i)} \left(\mathbb{E}[\eta_g(y^{(N)}, z^{(N)}, \alpha)] - \theta^{(i)}\right), \tag{101}$$

where $y^{(N)}$ and $z^{(N)}$ are the replicated datasets from the sampled values $y^{(i)}$ and $z^{(i)}$.

4.4.1 Stochastic Variational Inference for Gaussian Processes

To apply stochastic variational inference for Gaussian processes, we need a set of global parameters to optimize so that the optimization can be factored over the data points. The marginal likelihood which is usually optimized in Gaussian process framework cannot be factored over the training cases. To be able to factorize the objective function, Jensen's inequality is used in combination with the sparse GP framework (Hensman et al., 2013). Recall section 4.1 and the inducing variables $\mathbf{u}$ and assume that the observations $\mathbf{y}$ are Gaussian with mean $\mathbf{f}$ as in section 3.1.2. With the inducing variables, we have

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{f}) &= \mathrm{N}(\mathbf{y} \mid \mathbf{f}, \sigma^2 I), && (102) \\ p(\mathbf{f} \mid \mathbf{u}) &= \mathrm{N}(\mathbf{f} \mid \boldsymbol{\mu}, \tilde{K}), \\ &= \mathrm{N}(\mathbf{f} \mid K_{nu}K_{uu}^{-1}\mathbf{u}, K_{nn} - K_{nu}K_{uu}^{-1}K_{un}), && (103) \\ p(\mathbf{u}) &= \mathrm{N}(\mathbf{u} \mid \mathbf{0}, K_{uu}). && (104) \end{aligned}$$

By applying Jensen's inequality for conditional probability $p(\mathbf{y} \mid \mathbf{u})$, we can compute the lower bound of $\log p(\mathbf{y} \mid \mathbf{f})$:

$$\begin{aligned} \log p(\mathbf{y} \mid \mathbf{u}) &= \log \int p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{u}) d\mathbf{f}, \\ &\geq \int \log p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{u}) d\mathbf{f} = \mathcal{L}_1. && (105) \end{aligned}$$

Jensen's inequality is applied to $\log p(\mathbf{y} \mid \mathbf{f})$ because the computational complexity of computing $\log p(\mathbf{y} \mid \mathbf{u})$ analytically is $O(n^3)$ as there is the inversion of $n \times n$ matrix K_{nn}. By applying the Jensen's inequality, the computational complexity of computing the expectation is only $O(m^3)$ where m is the number of inducing variables $\mathbf{u}$. If we assume that the likelihood factorized over the data points as earlier, $\mathcal{L}_1$ can be computed analytically (see appenix) and results in

$$\exp(\mathcal{L}_1) = \prod_{i=1}^{n} \mathrm{N}(y_i \mid \mu_i, \sigma^2) \exp\left(-\frac{1}{2\sigma^2}\tilde{k}_{i,i}\right), \qquad (106)$$

where μ_i is the ith element of $\boldsymbol{\mu}$ and $\tilde{k}_{i.i}$ is the ith diagonal element of $\tilde{K}$.

As noted earlier, the inducing variables $\mathbf{u}$ work as the global latent variables in the formulation of stochastic variational inference in equations (80)–(88). In Gaussian process framework the local variables $\mathbf{z}$ are absent. By introducing variational distribution $q(\mathbf{u})$, the marginal likelihood can be bounded as in equation (65)

$$\begin{aligned} \log p(\mathbf{y} \mid X) &= \log \iint p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{u}, X) p(\mathbf{u}) d\mathbf{f} d\mathbf{u}, \\ &= \log \iint p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{u}, X) \frac{p(\mathbf{u})}{q(\mathbf{u})} q(\mathbf{u}) d\mathbf{f} d\mathbf{u}, \\ &\geq \int \left(\int \log p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{u}, X) d\mathbf{f} + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right) q(\mathbf{u}) d\mathbf{u}, \\ &= \int \left(\mathcal{L}_1 + \log p(\mathbf{u}) - \log q(\mathbf{u}) \right) q(\mathbf{u}) d\mathbf{u} = \mathcal{L}_2. && (107) \end{aligned}$$

The optimal variational distribution $q(\mathbf{u})$ is Gaussian based on the above. Using $q(\mathbf{u}) = \mathrm{N}(\mathbf{u} \mid \mathbf{m}, S)$, the bound $\mathcal{L}_2$ is

$$\mathcal{L}_2 = \sum_{i=1}^{n} \Big(\log \mathrm{N}(y_i \mid \mathbf{k}_i^T K_{uu}^{-1}\mathbf{m}, \sigma^2) - \frac{1}{2\sigma^2}\tilde{k}_{i,i} - \frac{1}{2}\mathrm{tr}(S\Lambda_i)\Big) - \mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})) \tag{108}$$

where $\mathbf{k}_i$ is the ith column of K_{uu} and $\Lambda_i = \frac{1}{\sigma^2}K_{uu}^{-1}\mathbf{k}_i\mathbf{k}_i^T K_{uu}^{-1}$. The KL divergence from $q(\mathbf{u}) = \mathrm{N}(\mathbf{u} \mid \mathbf{m}, S)$ to $p(\mathbf{u}) = \mathrm{N}(\mathbf{u} \mid \mathbf{0}, K_{uu})$ is

$$\mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})) = \frac{1}{2}\Big(\mathrm{tr}(K_{uu}^{-1}S) + \mathbf{m}^T K_{uu}^{-1}\mathbf{m} - m - \log\frac{|S|}{|K_{uu}|}\Big). \tag{109}$$

The marginal likelihood lower bound $\mathcal{L}_2$ is now factorized with respect to the observations. This means that the stochastic variational inference can be applied to Gaussian processes.

The gradients of the bound $\mathcal{L}_2$ with respect to the variational parameters are

$$\frac{\partial\mathcal{L}_2}{\partial\mathbf{m}} = \frac{1}{\sigma^2}K_{uu}^{-1}K_{un}\mathbf{y} - \Lambda\mathbf{m} \tag{110}$$

$$\frac{\partial\mathcal{L}_2}{\partial\mathbf{S}} = \frac{1}{2}S^{-1} - \frac{1}{2}\Lambda. \tag{111}$$

The stochastic variational inference works in the natural parameter space of the variational distribution. The mean and covariance of the variational distribution $q(\mathbf{u})$ can be converted to the natural parameters with

$$\boldsymbol{\theta}_1 = S^{-1}\mathbf{m} \tag{112}$$

$$\boldsymbol{\theta}_2 = -\frac{1}{2}S^{-1} \tag{113}$$

These natural parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ correspond to the global variational parameters λ in the previous sections. The natural gradients of the parameters are computed with

$$\hat{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_2 = G(\boldsymbol{\theta})^{-1}\frac{\partial\mathcal{L}_2}{\partial\boldsymbol{\theta}}, \tag{114}$$

where $G(\boldsymbol{\theta})$ is the Fisher Information matrix. For the exponential family distributions, we can utilize the *expectation* parameters

$$\boldsymbol{\eta}_1 = \mathbf{m}, \tag{115}$$

$$\boldsymbol{\eta}_2 = \mathbf{m}\mathbf{m}^T + S, \tag{116}$$

to compute the natural gradients. The natural parameters and expectation parameters are reciprocal, meaning that the gradient of one is the natural gradient of the other (Hensman et al., 2012, 2013)

$$\hat{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_2 = \nabla_{\boldsymbol{\eta}}\mathcal{L}_2, \qquad \hat{\nabla}_{\boldsymbol{\eta}}\mathcal{L}_2 = \nabla_{\boldsymbol{\theta}}\mathcal{L}_2. \tag{117}$$

Using (110) and (111), it is easy to see that the gradient of $\mathcal{L}_2$ with respect to the expectation parameters are equal to the gradient with respect to $\mathbf{m}$ and S. The natural parameters are thus updated with

$$\boldsymbol{\theta}_{1,(t+1)} = \boldsymbol{\theta}_{1,t} + \rho_t \left(\frac{1}{\sigma^2} K_{uu}^{-1} K_{un} \mathbf{y} - \Lambda \mathbf{m}_t \right), \tag{118}$$

$$\boldsymbol{\theta}_{2,(t+1)} = \boldsymbol{\theta}_{2,t} + \rho_t \left(\frac{1}{2} S_t^{-1} - \frac{1}{2} \Lambda \right). \tag{119}$$

With the bound $\mathcal{L}_2$ and its natural gradient with respect to the natural parameters of the variational distribution, the stochastic variational inference can be applied. As both the bound and the gradient can be decomposed to parts with respect to observations, we can apply the stochastic variational inference with either individual observations or minibatches of the data.

The variational distribution $q(\mathbf{u})$ approximates the true posterior distribution of the inducing inputs $p(\mathbf{u} \mid \mathbf{y}, X)$. The predictions with stochastic variational inference Gaussian processes (SVI-GP) are computed with

$$\begin{aligned} p(y^* \mid x^*, \mathbf{y}, X)) &= \int p(y^* \mid f^*) p(f^* \mid x^*, \mathbf{y}, X) df^*, \\ &= \iint p(y^* \mid f^*) p(f^* \mid x^*, \mathbf{u}) p(\mathbf{u} \mid \mathbf{y}, X) d\mathbf{u} df^*, \\ &\approx \iint p(y^* \mid f^*) p(f^* \mid x^*, \mathbf{u}) q(\mathbf{u}) d\mathbf{u} df^*. \end{aligned} \tag{120}$$

The likelihood and conditional distribution in (120) are analogous to (102) and (103)

$$p(y^* \mid f^*) = \mathrm{N}(y^* \mid f^*, \sigma^2), \tag{121}$$

$$p(f^* \mid x^*, \mathbf{u}) = \mathrm{N}(f^* \mid \mathbf{k}_{*u} K_{uu}^{-1} \mathbf{u}, k_{**} - \mathbf{k}_{*u} K_{uu}^{-1} \mathbf{k}_{u*}), \tag{122}$$

where $\mathbf{k}_{*u} = \mathbf{k}_{u*}^T$ is the covariance vector computed between the prediction input x^* and the inducing inputs $\mathbf{u}$ while k_{**} is the variance at the prediction input. Because all of the distributions in (120) are Gaussian, we can compute the predictions analytically

$$\begin{aligned} p(y^* \mid x^*, \mathbf{y}, X)) &\approx \iint p(y^* \mid f^*) p(f^* \mid x^*, \mathbf{u}) q(\mathbf{u}) d\mathbf{u} df^*, \\ &= \iint \mathrm{N}(y^* \mid f^*, \sigma^2) \mathrm{N}(f^* \mid \mathbf{k}_{*u} K_{uu}^{-1} \mathbf{u}, k_{**} - \mathbf{k}_{*u} K_{uu}^{-1} \mathbf{k}_{u*}) \mathrm{N}(\mathbf{u} \mid \mathbf{m}, S) d\mathbf{u} d\mathbf{f}^*, \\ &= \int \mathrm{N}(y^* \mid f^*, \sigma^2) \mathrm{N}(f^* \mid \mathbf{k}_{*u} K_{uu}^{-1} \mathbf{m}, k_{**} - \mathbf{k}_{*u} K_{uu}^{-1} \mathbf{k}_{u*} + \mathbf{k}_{*u} K_{uu}^{-1} S K_{uu}^{-1} \mathbf{k}_{u*}) df^*, \\ &= \mathrm{N}(y^* \mid \mathbf{k}_{*u} K_{uu}^{-1} \mathbf{m}, -\mathbf{k}_{*u} K_{uu}^{-1} \mathbf{k}_{u*} + \mathbf{k}_{*u} K_{uu}^{-1} S K_{uu}^{-1} \mathbf{k}_{u*} + \sigma^2). \end{aligned} \tag{123}$$

4.4.2 SVI-GP with non-Gaussian likelihood

In this thesis, we apply the stochastic variational inference to Gaussian process models with non-Gaussian likelihoods. Because the integral in the computation of

$\log p(\mathbf{y} \mid \mathbf{u})$ in (105) cannot be computed analytically if $p(\mathbf{y} \mid \mathbf{f})$ is not Gaussian, it is not possible to take such a straightforward approach as above. To be able to do the inference, we introduce *latent observations* and *latent likelihood* (Hensman et al., 2013). We assume that there is noise in the latent function $\mathbf{f}$ which produces the latent observations $\mathbf{y}$. Furthermore, we assume that this noise is normally distributed:

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathrm{N}(\mathbf{0}, \sigma^2 I). \tag{124}$$

Thus we treat the values $\mathbf{y}$ as the standard regression problem with Gaussian likelihood. The model likelihood then maps the latent observations to the true observations. The true observations are denoted as $\mathbf{t}$ in this case. The likelihood functions of the model can be summarized with

$$p(\mathbf{y} \mid \mathbf{f}) = \mathrm{N}(\mathbf{y} \mid \mathbf{f}, \sigma^2 I), \tag{125}$$

$$p(\mathbf{t} \mid \mathbf{f}) = \int p(\mathbf{t} \mid \mathbf{y}) p(\mathbf{y} \mid \mathbf{f}) d\mathbf{y}. \tag{126}$$

The use of latent likelihood enables the decomposition of the marginal likelihood to the Gaussian and non-Gaussian parts, which in turn enables the applying of stochastic variational inference to non-Gaussian likelihoods. The marginal likelihood is

$$p(\mathbf{t} \mid X) = \int p(\mathbf{t} \mid \mathbf{y}) \int p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid X) d\mathbf{f} d\mathbf{y} = \int p(\mathbf{t} \mid \mathbf{y}) p(\mathbf{y} \mid X) d\mathbf{y}. \tag{127}$$

The equation (127) has the Gaussian marginal likelihood part $p(\mathbf{y} \mid X)$ which is then combined with the actual likelihood $p(\mathbf{t} \mid \mathbf{y})$ to compute the real marginal likelihood. We assume the i.i.d. likelihood $p(\mathbf{t} \mid \mathbf{y}) = \prod i = 1^n p(t_i \mid y_i)$. Combining the likelihood with the Gaussian marginal likelihood bound $\mathcal{L}_2$ computed earlier and applying Jensen's inequality, the marginal likelihood is

$$\begin{aligned} p(\mathbf{t} \mid X) &= \log \int p(\mathbf{t} \mid \mathbf{y}) \int p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid X) d\mathbf{f} d\mathbf{y} = \int p(\mathbf{t} \mid \mathbf{y}) p(\mathbf{y} \mid X) d\mathbf{y}, \\ &\geq \int \exp\big(\mathcal{L}_2\big) p(\mathbf{t} \mid \mathbf{y}) d\mathbf{y}, \\ &= \int \exp\Big(\sum_{i=1}^{n} \big(\log \mathrm{N}(y_i \mid \mathbf{k}_i^T K_{uu}^{-1} \mathbf{m}, \sigma^2) - \frac{1}{2\sigma^2} \tilde{k}_{i,i} - \frac{1}{2} \mathrm{tr}(S\Lambda_i) \big) \\ &\quad - \mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})) \Big) p(\mathbf{t} \mid \mathbf{y}) d\mathbf{y}, \\ &= \prod_{i=1}^{n} \int \mathrm{N}(y_i \mid \mathbf{k}_i^T K_{uu}^{-1} \mathbf{m}, \sigma^2) p(t_i \mid y_i) dy_i \exp\Big(- \frac{1}{2\sigma^2} \tilde{k}_{i,i} - \frac{1}{2} \mathrm{tr}(S\Lambda_i) \\ &\quad - \mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})) \Big). \end{aligned} \tag{128}$$

In this thesis, the observations are either continuous with assumed Gaussian noise or binary variables. In the binary case, we use the *probit* likelihood

$$p(t_i \mid y_i) = \Phi(y_i t_i) = \int_{-\infty}^{y_i t_i} \mathrm{N}(h \mid 0, 1) dh. \tag{129}$$

The probit likelihood has the convenient quality in that it can be analytically integrated over Gaussian distribution (Rasmussen and Williams, 2006, pp. 74–75) which enables the analytic computation of $p(\mathbf{t} \mid X)$:

$$\int N(t \mid m, s^2)\Phi\left(\frac{t}{v}\right) dt = \begin{cases} \Phi(\frac{m}{\sqrt{v^2+s^2}}) & \text{if } v > 0 \\ \Phi(-\frac{m}{\sqrt{v^2+s^2}}) & \text{if } v < 0. \end{cases} \tag{130}$$

The marginal likelihood bound (128) can now be computed analytically. The log marginal likelihood bound becomes

$$\begin{aligned} \log p(\mathbf{t} \mid X) \geq & \sum_{i=1}^{n} \log\Big(\int \mathrm{N}(y_i \mid \mathbf{k}_i^T K_{uu}^{-1}\mathbf{m}, \sigma^2)\Phi(y_i t_i) dy_i\Big) \\ & - \frac{1}{2\sigma^2}\tilde{k}_{i,i} - \frac{1}{2}\mathrm{tr}(S\Lambda_i) - \mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})), \\ & = \sum_{i=1}^{n} \log\Phi(z_i) - \frac{1}{2\sigma^2}\tilde{k}_{i,i} - \frac{1}{2}\mathrm{tr}(S\Lambda_i) - \mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})) = \mathcal{L}_3, \end{aligned} \tag{131}$$

where

$$z_i = \frac{t_i \mathbf{k}_i^T K_{uu}^{-1}\mathbf{m}}{\sqrt{1+\sigma^2}}. \tag{132}$$

The predictions with non-Gaussian likelihood follow the Gaussian case. However, the integral over y^* in general cannot be computed and we have to resort to numerical integration. We decompose the predictive distribution $p(t^* \mid x^*, \mathbf{t}, X)$ again to the Gaussian and the non-Gaussian parts as we did with the marginal likelihood

$$p(t^* \mid x^*, \mathbf{t}, X) = \int p(t^* \mid y^*)p(y^* \mid x^*, \mathbf{t}, X)dy^*. \tag{133}$$

The Gaussian part $p(y^* \mid x^*, \mathbf{t}, X)$ can be computed as in (123), which we then combine with the likelihood $p(t^* \mid y^*)$ and integrate over y^*:

$$p(t^* \mid x^*, \mathbf{t}, X) = \int p(t^* \mid y^*)\mathrm{N}(y^* \mid \mu_*, \sigma_*^2)dy^*, \tag{134}$$

where

$$\mu_* = \mathbf{k}_{*u}K_{uu}^{-1}\mathbf{m}, \tag{135}$$

$$\sigma_*^2 = k_{**} - \mathbf{k}_{*u}K_{uu}^{-1}\mathbf{k}_{u*} + \mathbf{k}_{*u}K_{uu}^{-1}SK_{uu}^{-1}\mathbf{k}_{u*} + \sigma^2. \tag{136}$$

If we use the probit likelihood, predictive distribution (134) can be computed analytically as in (128)

$$\begin{aligned} p(t^* \mid x^*, \mathbf{t}, X) &= \int \Phi(y^* t^*)\mathrm{N}(y^* \mid \mu_*, \sigma_*^2)dy^*, \\ &= \Phi\left(t^* z_*\right), \end{aligned} \tag{137}$$

where

$$z_* = \frac{\mu_*}{\sqrt{1+\sigma_*^2}}. \tag{138}$$

Equation (137) computes the probability of the class $t^* \in \{-1, 1\}$ in binary classification. We can further summarize the predictive distribution with expected value and variance

$$\begin{aligned}
\mathbb{E}[t^*] &= \sum_{t^* \in \{-1,1\}} t^* p(t^* \mid x^*, \mathbf{t}, X) = \Phi\left(z_*\right) - \Phi\left(-z_*\right), \\
&= \Phi\left(z_*\right) - \left(1 - \Phi\left(z_*\right)\right) = 2\Phi\left(z_*\right) - 1, && (139)\\
\mathbb{E}[(t^*)^2] &= \sum_{t^* \in \{-1,1\}} (t^*)^2 p(t^* \mid x^*, \mathbf{t}, X) = \Phi\left(z_*\right) + \Phi\left(-z_*\right), \\
&= \Phi\left(z_*\right) + \left(1 - \Phi\left(z_*\right)\right) = 1, \\
\mathbb{V}[t^*] &= \mathbb{E}[(t^*)^2] - (\mathbb{E}[t^*])^2 = 1 - \left(2\Phi\left(z_*\right) - 1\right)^2, \\
&= -4\Phi\left(z_*\right)^2 + 4\Phi\left(z_*\right) = 4\Phi\left(z_*\right)\left(1 - \Phi\left(z_*\right)\right) && (140)
\end{aligned}$$

The variance of t^* in (140) is always positive because the cumulative Gaussian $\Phi \in (0, 1)$. The expected value (139) and the variance (140) are intuitive: the closer the cumulative Gaussian is to 1, the closer the expected value is to 1, and the smaller the variance becomes. Similarly, closer the cumulative Gaussian is to 0, the closer the expected value is to -1, and again, the smaller the variance is.

4.4.3 SVI-GP with monotonicity constraint

Now assume that we want to constrain the latent function $\mathbf{f}$ to be monotonic. By following the section 3.2, we use virtual observations t_v and $\mathbf{x}_v$ for the derivative of the latent function $\mathbf{f}'$ to denote that the latent function is either decreasing, $t_v = -1$, or increasing $t_v = 1$ in the virtual input $\mathbf{x}_v$. Following the above derivations, the SVI-GP framework is straightforward to include the monotonicity constraint. The setting now consists of the real observations $\{y_i, \mathbf{x}_i\}$; the virtual observations $\{t_{v,k}, \mathbf{x}_{v,k}\}$; and the inducing inputs $\{\mathbf{u}_j\}$, where $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n_v$ and $j = 1, 2, \ldots, m$. Using $\mathbf{y}_v$ as the latent virtual observations, we can decompose the marginal likelihood as

$$\begin{aligned}
&p(\mathbf{t}_v, \mathbf{y} \mid X, X_v) \\
&= \int p(\mathbf{t}_v \mid \mathbf{y}_v) \iiiint p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{y}_v \mid \mathbf{f}') p(\mathbf{f}, \mathbf{f}' \mid \mathbf{u}, X, X_v) p(\mathbf{u}) d\mathbf{u} d\mathbf{f} d\mathbf{f}' d\mathbf{y}_v. \quad (141)
\end{aligned}$$

Reusing the joint notation $\mathbf{f}_{\text{joint}}$ and $\mathbf{y}_{\text{joint}}$

$$\mathbf{f}_{\text{joint}} = \begin{bmatrix} \mathbf{f} \\ \mathbf{f}' \end{bmatrix}, \quad \mathbf{y}_{\text{joint}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{y}_v \end{bmatrix}, \tag{142}$$

the marginal likelihood can be compressed as

$$
\begin{aligned}
&p(\mathbf{t}_v, \mathbf{y} \mid X, X_v) \\
&= \int p(\mathbf{t}_v \mid \mathbf{y}_v) \iiint p(\mathbf{y}_{\text{joint}} \mid \mathbf{f}_{\text{joint}}) p(\mathbf{f}_{joint} \mid \mathbf{u}, X, X_v) p(\mathbf{u}) d\mathbf{u} d\mathbf{f}_{\text{joint}} d\mathbf{y}_v, \\
&= \int p(\mathbf{t}_v \mid \mathbf{y}_v) p(\mathbf{y}_{\text{joint}} \mid X, X_v) d\mathbf{y}_v. \qquad (143)
\end{aligned}
$$

Equation (143) is now analogous to the marginal likelihood in (128), except that now we only integrate over the latent virtual observations $\mathbf{y}_v$. The log marginal likelihood can thus be expressed as

$$
\begin{aligned}
\log p(\mathbf{y}, \mathbf{t}_v \mid X, X_v) \geq &\sum_{i=1}^{n} \left(\log \mathrm{N}(y_i \mid \mathbf{k}_i^T K_{uu}^{-1} \mathbf{m}, \sigma^2) - \frac{1}{2\sigma^2} \tilde{k}_{i,i} - \frac{1}{2} \mathrm{tr}(S\Lambda_i) \right) \\
&+ \sum_{k=1}^{n_v} \left(\log \left(\int \mathrm{N}(y_{v,k} \mid \mathbf{k}_{n+k}^T K_{uu}^{-1} \mathbf{m}, \sigma_v^2) \Phi(y_{v,k} t_{v,k}) dy_{v,k} \right) - \frac{1}{2\sigma_v^2} \tilde{k}_{n+k,n+k} \right. \\
&\left. - \frac{1}{2} \mathrm{tr}(S\Lambda_{v,k}) \right) - \mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})), \qquad (144)
\end{aligned}
$$

where $\mathbf{k}_i$ is now the ith column vector and $\tilde{k}_{i,i}$ is the ith diagonal of the covariance matrix of the joint latent vector $\mathbf{f}_{\text{joint}}$

$$
\mathbf{f}_{\text{joint}} = \begin{bmatrix} \mathbf{f} \\ \mathbf{f}' \end{bmatrix}, \quad \mathrm{Cov}(\mathbf{f}_{\text{joint}}) = K_{\text{joint}} = \begin{bmatrix} K_{nn} & K_{nd} \\ K_{dn} & K_{dd} \end{bmatrix} = \begin{bmatrix} K(X, X) & K_{(}X, X_v) \\ K(X_v, X) & K(X_v, X_v) \end{bmatrix} \qquad (145)
$$

and

$$
\Lambda_i = \frac{1}{\sigma^2} K_{uu}^{-1} \mathbf{k}_i \mathbf{k}_i^T K_{uu}^{-1}, \qquad \Lambda_{v,k} = \frac{1}{\sigma_v^2} K_{uu}^{-1} \mathbf{k}_{n+k} \mathbf{k}_{n+k}^T K_{uu}^{-1}, \qquad (146)
$$

with σ_v^2 being the variance of the latent likelihood of the virtual observations $p(y_{v,i} \mid f_i') = \mathrm{N}(y_{v,i} \mid f_i', \sigma_v^2)$. The integrals can again be computed analytically

$$
\begin{aligned}
\log p(\mathbf{y}, \mathbf{t}_v \mid X, X_v) \geq &\sum_{i=1}^{n} \left(\log \mathrm{N}(y_i \mid \mathbf{k}_i^T K_{uu}^{-1} \mathbf{m}, \sigma^2) - \frac{1}{2\sigma^2} \tilde{k}_{i,i} - \frac{1}{2} \mathrm{tr}(S\Lambda_i) \right) \\
&+ \sum_{k=1}^{n_v} \left(\log \Phi \left(\frac{t_{v,k} \mathbf{k}_{n+k}^T K_{uu}^{-1} \mathbf{m}}{\sqrt{1 + \sigma_v^2}} \right) - \frac{1}{2\sigma_v^2} \tilde{k}_{n+k,n+k} - \frac{1}{2} \mathrm{tr}(S\Lambda_{v,k}) \right) \\
&- \mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})). \qquad (147)
\end{aligned}
$$

If the real observations are also binary variables as in section 4.4.2, log marginal likelihood bound becomes

$$\begin{aligned}
&\log p(\mathbf{t}, \mathbf{t}_v \mid X, X_v) \\
&\quad \geq \sum_{i=1}^{n} \left(\log \left(\int \mathrm{N}(y_i \mid \mathbf{k}_i^T K_{uu}^{-1} \mathbf{m}, \sigma^2) \Phi(y_i t_i) dy_i \right) - \frac{1}{2\sigma^2} \tilde{k}_{i,i} - \frac{1}{2} \mathrm{tr}(S \Lambda_i) \right) \\
&\qquad + \sum_{k=1}^{n_v} \left(\log \left(\int \mathrm{N}(y_{v,k} \mid \mathbf{k}_{n+k}^T K_{uu}^{-1} \mathbf{m}, \sigma_v^2) \Phi(y_{v,k} t_{v,k}) dy_{v,k} \right) \right. \\
&\qquad \left. - \frac{1}{2\sigma_v^2} \tilde{k}_{n+k,n+k} - \frac{1}{2} \mathrm{tr}(S \Lambda_{v,k}) \right) - \mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})), \\
&\quad = \sum_{i=1}^{n} \left(\log \Phi \left(\frac{t_i \mathbf{k}_i^T K_{uu}^{-1} \mathbf{m}}{\sqrt{1+\sigma^2}} \right) - \frac{1}{2\sigma^2} \tilde{k}_{i,i} - \frac{1}{2} \mathrm{tr}(S \Lambda_i) \right) + \sum_{k=1}^{n_v} \left(\log \Phi \left(\frac{t_{v,k} \mathbf{k}_{n+k}^T K_{uu}^{-1} \mathbf{m}}{\sqrt{1+\sigma_v^2}} \right) \right. \\
&\qquad \left. - \frac{1}{2\sigma_v^2} \tilde{k}_{n+k,n+k} - \frac{1}{2} \mathrm{tr}(S \Lambda_{v,k}) \right) - \mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})). \qquad (148)
\end{aligned}$$

5 Parameter estimation and integration

The parameters that define how the function values $\mathbf{f}$ behave a prior are the parameters of the covariance functions (hyperparameters) and the noise variance σ^2 in the likelihood function. In this section, we go over the methods we use to set these parameters.

5.1 Maximum a posteriori estimate

Lets denote with θ the parameters of the GP model, including the hyperparameters and the likelihood parameters. In section 2, we went over the basic of Bayesian modelling. Using the earlier notation of observations $\mathbf{y}$ with inputs X, the posterior density (1) of the parameters is

$$p(\theta \mid \mathbf{y}, X) = \frac{p(\mathbf{y} \mid \theta, X)p(\theta)}{p(\mathbf{y} \mid X)} \propto p(\mathbf{y} \mid \theta, X)p(\theta), \tag{149}$$

where $p(\mathbf{y} \mid X)$ does not depend on θ. In GP framework, the likelihood of the observations $p(\mathbf{y} \mid \theta, X)$ is actually the marginal likelihood, where we have marginalized, or integrated over, the latent values $\mathbf{f}$. Here we have explicitly expressed the conditioning on the parameters θ, which is often left out for notational convenience. To do the full Bayesian analysis, we would have to use the posterior density for θ and do the inference with it. In this thesis we adopt the use of MAP point estimate for the parameters, due to computational reasons and convenience. The MAP values correspond to maximum a posteriori values, the values that maximize the corresponding posterior density.

The MAP values for the parameters can be found by maximizing the posterior density (149) with respect to the parameters. We utilize gradient based methods in order to find the optimal parameter values. Gradient based methods use the gradients of the objective function iteratively to find the maximum or the minimum. It is common practice to maximize the logarithm of the (un)normalized posterior density instead of the normalized posterior density. Because the logarithm is a monotonic function, the maximum of the posterior density is the same as the maximum of the logarithm of the posterior density. Furthermore, the logarithm can make the objective function resemble quadratic function which is better suited for gradient methods.

We already encountered one class of gradient methods, namely the stochastic gradient ascend, when going over the stochastic variational inference in section 4.4. The gradient ascend method finds the optimum parameter value by taking successive steps towards the direction of the gradient

$$\theta_{i+1} = \theta_i + \rho g_i = \theta_i + \rho \frac{\partial \mathcal{L}(\theta_i)}{\partial \theta}, \tag{150}$$

where ρ is the step size and $\mathcal{L}$ is the objective function to be maximized. While being intuitive, the standard gradient ascend method suffers from several drawbacks. The major drawback of the gradient ascend is that the convergence to the optimum can be very slow near the maximum or minimum. The gradient ascend can also suffer from oscillating updates due to the step-size being too large. Due to the limitations of the gradient ascend, we use conjugate gradient methods in this thesis.

5.1.1 Conjugate Gradient methods

Conjugate gradient methods (Hestenes and Stiefel, 1952) are methods utilizing gradients for finding the minimum or maximum of function $f(x)$. The conjuagate gradient methods work by taking steps towards the *conjugate* direction which is a linear combination of the steepest (gradient) direction and the direction of the previous conjugate step. In contrast to standard gradient ascend method, the conjugate gradient methods have better convergence properties and behaviour.

The conjugate gradient method for finding the maximum of general nonlinear function $f(x)$ can be summarized as

1. Compute the gradient $g_i = \nabla_x f(x_i)$ in the current point x_i.
2. Compute the conjugate coefficient β_i.
3. Update the conjugate direction $s_i = g_i + \beta_i s_{i-1}$.
4. Perform a line search on s_i: Find the maximum of the function $f(x_i + \alpha s_i)$ with respect to α.
5. Update the current point $x_{i+1} = x_i + \alpha s_i$.

The conjugate gradient algorithm is initialized with $s_0 = 0$, so the first step is always towards the direction of the gradient. There are several possible methods for computing the conjugate coefficient β_i. The most popular are the Flethcher-Reeves (FR), Polar-Ribière (PR), and Hestenes-Stiefel (HS):

$$\beta_i^{\mathrm{FR}} = \frac{g_i^T g_i}{g_{i-1}^T g_{i-1}}, \tag{151}$$

$$\beta_i^{\mathrm{PR}} = \frac{g_i^T (g_i - g_{i-1})}{g_{i-1}^T g_{i-1}}, \tag{152}$$

$$\beta_i^{\mathrm{HS}} = -\frac{g_i^T (g_i - g_{i-1})}{s_{i-1}^T (g_i - g_{i-1})}. \tag{153}$$

In this thesis, we use the Polar-Ribière conjugate coefficient β^{PR}. We also reset the conjugate direction s_i to point towards the direction of the gradient after we have taken n_p steps, where n_p is the number of parameters to be optimized.

5.1.2 Gradients of the marginal likelihoods

To optimize the parameters of the GP model, we need the gradient of the marginal likelihood with respect to the hyperparameters and the likelihood parameters. To recap, the log marginal likelihood in the GP regression is

$$\log p(\mathbf{y} \mid \theta, \sigma^2, X) = -\frac{1}{2}\mathbf{y}^T(K_\theta + \sigma^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log|K_\theta + \sigma^2 I| - \frac{n}{2}\log 2\pi, \tag{154}$$

where we have explicitly denoted the conditioning on hyperparameters θ and the noise variance σ^2. The gradient of (154) with respect to the hyperparameters θ and the noise variance σ^2 is

$$\begin{aligned} \frac{\partial \log p(\mathbf{y} \mid \theta, \sigma^2, X)}{\partial \theta} &= \frac{1}{2}\mathbf{y}^T(K_\theta + \sigma^2 I)^{-1}\frac{\partial K_\theta}{\partial \theta}(K_\theta + \sigma^2 I)^{-1}\mathbf{y} \\ &\quad - \frac{1}{2}\mathrm{tr}\left((K_\theta + \sigma^2 I)^{-1}\frac{\partial K_\theta}{\partial \theta}\right), &(155) \\ \frac{\partial \log p(\mathbf{y} \mid \theta, \sigma^2, X)}{\partial \sigma^2} &= \frac{1}{2}\mathbf{y}^T(K_\theta + \sigma^2 I)^{-1}(K_\theta + \sigma^2 I)^{-1}\mathbf{y} \\ &\quad - \frac{1}{2}\mathrm{tr}((K_\theta + \sigma^2 I)^{-1}). &(156) \end{aligned}$$

The EP approximation to the marginal likelihood in the case of non-Gaussian likelihood is

$$\begin{aligned} \log Z_{\mathrm{EP}} = \sum_{i=1}^{n} \log \hat{Z}_i + \frac{1}{2}\log(\tilde{\sigma}_i^2 + \sigma_{-i}^2) + \frac{(\tilde{\mu}_i - \mu_{-i})^2}{2(\tilde{\sigma}_i^2 + \sigma_{-i}^2)} \\ - \frac{1}{2}\log|K_\theta + \tilde{\Sigma}| - \frac{1}{2}\tilde{\boldsymbol{\mu}}^T(K_\theta + \tilde{\Sigma})^{-1}\tilde{\boldsymbol{\mu}}. \end{aligned} \tag{157}$$

If the EP algorithm has converged, the implicit derivatives of the EP site approximations with respect to the hyperparameters are exactly zero (Seeger, 2005). Thus, the gradient of the EP approximation to the log marginal likelihood is

$$\frac{\partial \log Z_{\mathrm{EP}}}{\partial \theta} = -\frac{1}{2}\mathrm{tr}\left((K_\theta + \tilde{\Sigma})^{-1}\frac{\partial K_\theta}{\partial \theta}\right) + \frac{1}{2}\tilde{\boldsymbol{\mu}}^T(K_\theta + \tilde{\Sigma})^{-1}\frac{\partial K_\theta}{\partial \theta}(K_\theta + \tilde{\Sigma})^{-1}\tilde{\boldsymbol{\mu}}. \tag{158}$$

For the monotonic GP, we have a combination of the Gaussian process regression, with the EP classification of the virtual observations. The log marginal likelihood is

$$\begin{aligned} \log Z_{\mathrm{EP,m}} = \sum_{i=1}^{m}\left(\log \hat{Z}_i + \frac{1}{2}\log(\tilde{\sigma}_i^2 + \sigma_{-i}^2) + \frac{(\tilde{\mu}_i - \mu_{-i})^2}{2(\tilde{\sigma}_i^2 + \sigma_{-i}^2)}\right) + \frac{m-n}{2}\log 2\pi \quad &(159) \\ - \frac{1}{2}\log|K_{joint} + \tilde{\Sigma}_{joint}| - \frac{1}{2}\tilde{\boldsymbol{\mu}}_{joint}^T(K_{joint} + \tilde{\Sigma}_{joint})^{-1}\tilde{\boldsymbol{\mu}}_{joint}. \quad &(160) \end{aligned}$$

The derivatives of the EP site approximation terms are again zero, with respect to the hyperparameters of the model. However, here we have the likelihood noise σ^2 in addition to the hyperparameters in the joint site covariance matrix

$$\tilde{\Sigma}_{joint} = \begin{bmatrix} \sigma^2 I & 0 \\ 0 & \tilde{\Sigma} \end{bmatrix}. \tag{161}$$

The gradient with respect to the hyperparameters is equal to (158) and the gradient with respect to the noise variance σ^2 is

$$\frac{\partial \log Z_{\mathrm{EP,m}}}{\partial \sigma^2} = -\frac{1}{2}\mathrm{tr}\left((K_\theta + \tilde{\Sigma}_{joint})^{-1} \frac{\partial \tilde{\Sigma}_{joint}}{\partial \sigma^2} \right) + \frac{1}{2}\tilde{\boldsymbol{\mu}}^T (K_\theta + \tilde{\Sigma}_{joint})^{-1} \frac{\partial \tilde{\Sigma}_{joint}}{\partial \sigma^2} (K_\theta + \tilde{\Sigma}_{joint})^{-1} \tilde{\boldsymbol{\mu}}, \tag{162}$$

where

$$\frac{\partial \tilde{\Sigma}_{joint}}{\partial \sigma^2} = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{163}$$

For the stochastic variational inference with monotonicity constraint, where the observations are continous and we assume the Gaussian likelihood $p(y_i \mid f_i) = \mathrm{N}(y_i, \mid f_i, \sigma^2)$, the lower bound to the log marginal likelihood is

$$\log p(\mathbf{y}, \mathbf{t}_v \mid X, X_v) \geq \sum_{i=1}^{n} \left(\log \mathrm{N}(y_i \mid \mathbf{k}_i^T K_{uu}^{-1} \mathbf{m}, \sigma^2) - \frac{1}{2\sigma^2}\tilde{k}_{i,i} - \frac{1}{2}\mathrm{tr}(S\Lambda_i) \right) + \sum_{k=1}^{n_v} \left(\log \Phi \left(\frac{t_{v,k} \mathbf{k}_{n+k}^T K_{uu}^{-1} \mathbf{m}}{\sqrt{1+\sigma_v^2}} \right) - \frac{1}{2\sigma_v^2} \tilde{k}_{n+k,n+k} - \frac{1}{2}\mathrm{tr}(S\Lambda_{v,k}) \right) - \mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})) = \mathcal{L}_4. \tag{164}$$

The gradients with respect to θ can be realized with the help of the chain derivation rule

$$\frac{\partial \Phi(z_i)}{\partial \theta} = \frac{\partial \Phi(z_i)}{\partial z_i}\frac{\partial z_i}{\partial \theta} = \mathrm{N}(z_i \mid 0, 1)\frac{\partial z_i}{\partial \theta}. \tag{165}$$

To simplify the notation, we compute the gradients of the individual terms. First the log Gaussian terms

$$\frac{\partial \log \mathrm{N}(y_i \mid \mathbf{k}_i^T K_{uu}^{-1}\mathbf{m}, \sigma^2)}{\partial \theta} = \frac{\partial}{\partial \theta}\left(-\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(y_i - \mathbf{k}_i^T K_{uu}^{-1}\mathbf{m}\right)^2 \right), = \frac{1}{\sigma^2}\left(y_i - \mathbf{k}_i^T K_{uu}^{-1}\mathbf{m}\right)\left(\left(\frac{\partial \mathbf{k}_i}{\partial \theta}\right)^T K_{uu}^{-1}\mathbf{m} - \mathbf{k}_i^T K_{uu}^{-1}\frac{\partial K_{uu}}{\partial \theta}K_{uu}^{-1}\mathbf{m} \right), \tag{166}$$

$$\frac{\partial \log \mathrm{N}(y_i \mid \mathbf{k}_i^T K_{uu}^{-1}\mathbf{m}, \sigma^2)}{\partial \sigma^2} = -\frac{1}{2\sigma^2} + \frac{1}{2\sigma^4}\left(y_i - \mathbf{k}_i^T K_{uu}^{-1}\mathbf{m}\right)^2. \tag{167}$$

The gradients with respect to the log cumulative Gaussian Φ are

$$\frac{\partial}{\partial \theta}\log \Phi\left(z_i\right) = \Phi\left(z_i\right)^{-1}\frac{\partial}{\partial \theta}\Phi\left(z_i\right) = \Phi\left(z_i\right)^{-1}\mathrm{N}\left(z_i \mid 0, 1\right)\frac{\partial z_i}{\partial \theta}, \tag{168}$$

$$\frac{\partial}{\partial \sigma_v^2}\log \Phi\left(z_i\right) = \Phi\left(z_i\right)^{-1}\mathrm{N}\left(z_i \mid 0, 1\right)\frac{\partial z_i}{\partial \sigma_v^2}, \tag{169}$$

where the z_i and its gradients are

$$
\begin{align}
z_i &= \frac{t_{v,k}\mathbf{k}_{n+k}^T K_{uu}^{-1}\mathbf{m}}{\sqrt{1+\sigma_v^2}}, \tag{170}\\
\frac{\partial z_i}{\partial \theta} &= \frac{t_{v,k}}{\sqrt{1+\sigma_v^2}} \left(\left(\frac{\partial \mathbf{k}_{n+k}}{\partial \theta} \right)^T K_{uu}^{-1}\mathbf{m} - \mathbf{k}_{n+k}^T K_{uu}^{-1} \frac{\partial K_{uu}}{\partial \theta} K_{uu}^{-1}\mathbf{m} \right), \tag{171}\\
\frac{\partial z_i}{\partial \sigma_v^2} &= -\frac{t_{v,k}\mathbf{k}_{n+k}^T K_{uu}^{-1}\mathbf{m}}{2(1+\sigma_v^2)^{3/2}}. \tag{172}
\end{align}
$$

The trace terms are linear, so the gradients can be computed as

$$
\begin{align}
\frac{\partial \mathrm{tr}(S\Lambda_i)}{\partial \theta} &= \mathrm{tr}\left(S \frac{\partial}{\partial \theta} \left(\frac{1}{\sigma^2} K_{uu}^{-1}\mathbf{k}_i\mathbf{k}_i^T K_{uu}^{-1} \right) \right), \nonumber\\
&= \mathrm{tr}\left(S \left(-\frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial K_{uu}}{\partial \theta} K_{uu}^{-1}\mathbf{k}_i\mathbf{k}_i^T K_{uu}^{-1} + \frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial \mathbf{k}_i}{\partial \theta} \mathbf{k}_i^T K_{uu}^{-1} \right) \right), \tag{173}\\
\frac{\partial \mathrm{tr}(S\Lambda_i)}{\partial \sigma^2} &= \mathrm{tr}\left(S \frac{\partial}{\partial \sigma^2} \left(\frac{1}{\sigma^2} K_{uu}^{-1}\mathbf{k}_i\mathbf{k}_i^T K_{uu}^{-1} \right) \right) = \mathrm{tr}\left(-\frac{1}{\sigma^2} S\Lambda_i \right), \tag{174}\\
\frac{\partial \mathrm{tr}(S\Lambda_{v,k})}{\partial \theta} &= \mathrm{tr}\left(S \frac{\partial}{\partial \theta} \left(\frac{1}{\sigma_v^2} K_{uu}^{-1}\mathbf{k}_{n+k}\mathbf{k}_{n+k}^T K_{uu}^{-1} \right) \right), \nonumber\\
&= \mathrm{tr}\left(S \left(-\frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial K_{uu}}{\partial \theta} K_{uu}^{-1}\mathbf{k}_{n+k}\mathbf{k}_{n+k}^T K_{uu}^{-1} + \frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial \mathbf{k}_{n+k}}{\partial \theta} \mathbf{k}_{n+k}^T K_{uu}^{-1} \right) \right), \tag{175}\\
\frac{\partial \mathrm{tr}(S\Lambda_{v,k})}{\partial \sigma_v^2} &= \mathrm{tr}\left(-\frac{1}{\sigma_v^2} S\Lambda_{v,k} \right). \tag{176}
\end{align}
$$

The gradient of the KL divergence between the two Gaussians is

$$
\begin{align}
\frac{\partial \mathrm{KL}(p(\mathbf{u}) \parallel q(\mathbf{u}))}{\partial \theta} &= \frac{\partial}{\partial \theta} \left(\frac{1}{2} \left(\mathrm{tr}(K_{uu}^{-1}S) + \mathbf{m}^T K_{uu}^{-1}\mathbf{m} - m - \log \frac{|S|}{|K_{uu}|} \right) \right), \nonumber\\
&= -\frac{1}{2}\mathrm{tr}\left(K_{uu}^{-1} \frac{\partial K_{uu}}{\partial \theta} K_{uu}^{-1} S \right) - \mathbf{m}^T K_{uu}^{-1} \frac{\partial K_{uu}}{\partial \theta} K_{uu}^{-1}\mathbf{m} + \mathrm{tr}\left(K_{uu}^{-1} \frac{\partial K_{uu}}{\partial \theta} \right). \tag{177}
\end{align}
$$

Using (166)–(177), the gradients of $\mathcal{L}_4$ are

$$
\begin{align}
\frac{\partial \mathcal{L}_4}{\partial \theta} = \sum_{i=1}^{n} \Bigg(& \frac{1}{\sigma^2} \left(y_i - \mathbf{k}_i^T K_{uu}^{-1}\mathbf{m} \right) \left(\left(\frac{\partial \mathbf{k}_i}{\partial \theta} \right)^T K_{uu}^{-1}\mathbf{m} - \mathbf{k}_i^T K_{uu}^{-1} \frac{\partial K_{uu}}{\partial \theta} K_{uu}^{-1}, \mathbf{m} \right) \nonumber\\
& - \frac{1}{2\sigma^2} \frac{\partial \tilde{k}_{i,i}}{\partial \theta} - \frac{1}{2}\mathrm{tr}\left(S \left(-\frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial K_{uu}}{\partial \theta} K_{uu}^{-1}\mathbf{k}_i\mathbf{k}_i^T K_{uu}^{-1} + \frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial \mathbf{k}_i}{\partial \theta} \mathbf{k}_i^T K_{uu}^{-1} \right) \right) \Bigg) \nonumber\\
+ \sum_{k=1}^{n_v} \Bigg(& \Phi\left(z_i\right)^{-1} \mathrm{N}\left(z_i \mid 0, 1\right) \frac{\partial z_i}{\partial \theta} - \frac{1}{2\sigma_v^2} \frac{\partial \tilde{k}_{n+k,n+k}}{\partial \theta} \nonumber\\
& - \frac{1}{2}\mathrm{tr}\left(S \left(-\frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial K_{uu}}{\partial \theta} K_{uu}^{-1}\mathbf{k}_{n+k}\mathbf{k}_{n+k}^T K_{uu}^{-1} + \frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial \mathbf{k}_{n+k}}{\partial \theta} \mathbf{k}_{n+k}^T K_{uu}^{-1} \right) \right) \Bigg), \tag{178}
\end{align}
$$

$$\frac{\partial \mathcal{L}_4}{\partial \sigma^2} = \sum_{i=1}^{n} \left(-\frac{1}{2\sigma^2} + \frac{1}{2\sigma^4} \left(y_i - \mathbf{k}_i^T K_{uu}^{-1} \mathbf{m} \right)^2 + \frac{1}{2\sigma^4} \tilde{k}_{i,i} + \operatorname{tr}\left(\frac{1}{\sigma^2} S \Lambda_i \right) \right), \tag{179}$$

$$\frac{\partial \mathcal{L}_4}{\partial \sigma_v^2} = \sum_{k=1}^{n_v} \left(\Phi\left(z_i\right)^{-1} \mathrm{N}\left(z_i \mid 0, 1\right) \frac{\partial z_i}{\partial \sigma_v^2} - \frac{1}{2\sigma_v^4} \tilde{k}_{n+k,n+k} + \operatorname{tr}\left(\frac{1}{\sigma_v^2} S \Lambda_{v,k} \right) \right), \tag{180}$$

where z_i and its gradients are computed with (170)–(172).

If the real observations are also binary variables, the lower bound to the marginal likelihood is

$$\begin{aligned} \log p(\mathbf{t}, \mathbf{t}_v \mid X, X_v) \geq \sum_{i=1}^{n} \Big(\log \Phi\left(z_{1,i}\right) - \frac{1}{2\sigma^2} \tilde{k}_{i,i} - \frac{1}{2} \operatorname{tr}(S \Lambda_i) \Big) + \sum_{k=1}^{n_v} \Big(\log \Phi\left(z_{2,k}\right) \\ - \frac{1}{2\sigma_v^2} \tilde{k}_{n+k,n+k} - \frac{1}{2} \operatorname{tr}(S \Lambda_{v,k}) \Big) - \mathrm{KL}(q(\mathbf{u}) \parallel p(\mathbf{u})) = \mathcal{L}_5, \end{aligned} \tag{181}$$

where

$$z_{1,i} = \frac{t_i \mathbf{k}_i^T K_{uu}^{-1} \mathbf{m}}{\sqrt{1 + \sigma^2}}, \qquad z_{2,k} = \frac{t_{v,k} \mathbf{k}_{n+k}^T K_{uu}^{-1} \mathbf{m}}{\sqrt{1 + \sigma_v^2}}. \tag{182}$$

The only difference between the log marginal likelihood bounds $\mathcal{L}_4$ in (164) and $\mathcal{L}_5$ in (181) is that the log Gaussian term $\log \mathrm{N}(\cdot)$ is now replaced by the log cumulative Gaussian $\log \Phi(\cdot)$. Thus, the gradients can be computed by replacing the gradient of the log Gaussian with the gradient of the log cumulative Gaussian. The gradient of the log cumulative Gaussian, $\log \Phi(z_{1,i})$, can be computed analogously to the gradient of $\log \Phi(z_{2,i})$ which has been computed already in (168)–(172). The gradients of $\mathcal{L}_5$ are

$$\begin{aligned} \frac{\partial \mathcal{L}_5}{\partial \theta} = \sum_{i=1}^{n} \Bigg(& \Phi\left(z_{1,i}\right)^{-1} \mathrm{N}\left(z_{1,i} \mid 0, 1\right) \frac{\partial z_{1,i}}{\partial \theta} - \frac{1}{2\sigma^2} \frac{\partial \tilde{k}_{i,i}}{\partial \theta} \\ & - \frac{1}{2} \operatorname{tr}\left(S \left(-\frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial K_{uu}}{\partial \theta} K_{uu}^{-1} \mathbf{k}_i \mathbf{k}_i^T K_{uu}^{-1} + \frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial \mathbf{k}_i}{\partial \theta} \mathbf{k}_i^T K_{uu}^{-1} \right) \right) \Bigg) \\ & + \sum_{k=1}^{n_v} \Bigg(\Phi\left(z_{2,k}\right)^{-1} \mathrm{N}\left(z_{2,k} \mid 0, 1\right) \frac{\partial z_{2,k}}{\partial \theta} - \frac{1}{2\sigma_v^2} \frac{\partial \tilde{k}_{n+k,n+k}}{\partial \theta} \\ & - \frac{1}{2} \operatorname{tr}\left(S \left(-\frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial K_{uu}}{\partial \theta} K_{uu}^{-1} \mathbf{k}_{n+k} \mathbf{k}_{n+k}^T K_{uu}^{-1} + \frac{2}{\sigma^2} K_{uu}^{-1} \frac{\partial \mathbf{k}_{n+k}}{\partial \theta} \mathbf{k}_{n+k}^T K_{uu}^{-1} \right) \right) \Bigg), \end{aligned} \tag{183}$$

$$\frac{\partial \mathcal{L}_5}{\partial \sigma^2} = \sum_{k=1}^{n_v} \left(\Phi\left(z_{1,i}\right)^{-1} \mathrm{N}\left(z_{1,i} \mid 0, 1\right) \frac{\partial z_{1,i}}{\partial \sigma_v^2} - \frac{1}{2\sigma_v^4} \tilde{k}_{n+k,n+k} + \operatorname{tr}\left(\frac{1}{\sigma_v^2} S \Lambda_{v,k} \right) \right), \tag{184}$$

$$\frac{\partial \mathcal{L}_5}{\partial \sigma_v^2} = \sum_{k=1}^{n_v} \left(\Phi\left(z_{2,k}\right)^{-1} \mathrm{N}\left(z_{2,k} \mid 0, 1\right) \frac{\partial z_{2,k}}{\partial \sigma_v^2} - \frac{1}{2\sigma_v^4} \tilde{k}_{n+k,n+k} + \operatorname{tr}\left(\frac{1}{\sigma_v^2} S \Lambda_{v,k} \right) \right). \tag{185}$$

5.2 Markov Chain Monte Carlo

The MAP estimates for the parameters are usually good enough for practical application. However, these point estimates for the parameters discard the uncertainty of the parameters. If we want to do full Bayesian inference and integrate over the uncertainty of the unknown parameters, the distributions and inference quickly become intractable. This is often the case with complex statistical models.

Monte Carlo methods are sampling based methods where instead of using exact distributions in the inference, we sample the appropriate distributions to get samples that represent the distribution of interest. One-dimensional distributions can be easily sampled by computing the distribution in a grid and using for example the inverse transform sampling. However, when sampling multiple parameters at the same time, tabulating the distribution values is no longer a viable option. If there are multiple parameters and the functional form of the joint distribution of the parameters is very complex, and cannot be factored to easy to sample conditional distributions, getting true samples from the joint distribution is usually impossible.

Markov Chain Monte Carlo (MCMC, Metropolis and Ulam, 1949; Metropolis et al., 1953; Hastings, 1970) methods are extensions to standard Monte Carlo methods which enable the sampling of arbitrary distributions. The downside of the MCMC methods is that the representative samples are now correlated and thus are not real Monte Carlo samples from the distribution. The fact that the samples correlate means that in order to get the full picture of the distribution, more samples are needed than with standard Monte Carlo methods.

If we want to do full Bayesian analysis with Gaussian processes (Neal, 1997), we need to infer the posterior distribution of the latent function values $\mathbf{f}$ and the parameters θ of the GP model

$$p(\theta, \mathbf{f} \mid \mathbf{y}, X) = p(\mathbf{f} \mid \theta, \mathbf{y}, X) p(\theta \mid \mathbf{y}, X). \tag{186}$$

For Gaussian likelihood, the posterior distribution $p(\mathbf{f} \mid \theta, \mathbf{y}, X)$ is analytically tractable and therefore we only need to sample the parameters θ (see section 3). However, if the posterior distribution of $\mathbf{f}$ is not analytically tractable, the posterior of the latent values needs to be either approximated, for example with EP, or sampled. To sample the posterior $p(\theta, \mathbf{f} \mid \mathbf{y}, X)$, we alternate between the sampling of $p(\mathbf{f} \mid \theta, \mathbf{y}, X)$ and $p(\theta \mid \mathbf{f}, X)$. Note that if $\mathbf{f}$ is known, the dependency on $\mathbf{y}$ can be dropped. The general scheme for full Bayesian analysis with Gaussian processes is as follows

Full MCMC for Gaussian processes:

1. Initialize $\mathbf{f}$ and θ to some appropriate values.

2. Given the most recent sample for θ, sample $\mathbf{f}$ from its posterior distribution

$$p(\mathbf{f} \mid \theta, \mathbf{y}, X) = \frac{p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \theta, X)}{p(\mathbf{y} \mid \theta, X)}. \tag{187}$$

3. Given the most recent sample for $\mathbf{f}$, sample θ from its posterior

$$p(\theta \mid \mathbf{f}, X) = \frac{p(\mathbf{f} \mid \theta, X) p(\theta)}{p(\mathbf{f} \mid X)} \propto \mathrm{N}(\mathbf{f} \mid \mathbf{0}, K_\theta) p(\theta). \tag{188}$$

4. If not enough samples, go to 2., otherwise end.

It is also possible to do *partial* or *latent* MCMC where we only sample the parameters θ. In this case the marginal likelihood $p(\mathbf{f} \mid \theta, \mathbf{y}, X)$ is approximated with some suitable method, for example with EP. The hyperparameters are then sampled straight from the posterior approximation $p(\theta \mid \mathbf{y}, X)$

Latent MCMC for Gaussian processes:

1. Initialize θ to appropriate values.
2. Approximate the posterior of θ using suitable method, for example EP

$$p(\theta \mid \mathbf{y}, X) \propto p(\mathbf{y} \mid \theta, X) p(\theta) \approx q(\mathbf{y} \mid X) p(\theta), \tag{189}$$

   where $q(\mathbf{y} \mid X)$ is the approximation to the marginal likelihood $p(\mathbf{y} \mid X)$, for example (157).
3. Sample the posterior of θ
4. If not enough samples, go to 2., otherwise end.

In this thesis, we use both the full MCMC and the latent MCMC and compare these to methods using point estimates. The parameters θ are sampled using slice sampling (Neal, 2003) and the latent values $\mathbf{f}$ are sampled using elliptical slice sampling (Murray et al., 2009).

5.2.1 Slice sampling

In this thesis, we use slice sampling (Neal, 2003) to sample the parameters of the covariance and likelihood function. In slice sampling the next value for the parameter is sampled uniformly from the interval defined by the previous *slice*. The slice is the density of the sampled distribution at the current sample. This slice is then sampled uniformly to get the limits for sampling of the next values for the parameters.

Figure 2 displays the procedure. The vertical line is the slice defined at z_i from 0 to $\tilde{p}(z_i)$ where $\tilde{p}$ is the distribution which we wish to sample (normalized or unnormalized). The random point u is then randomly sampled from interval $[0, \tilde{p}(z_i)]$. The variable u defines the interval which we sample for the next sample of the parameters. Our aim is to get the next sample from the interval which is defined by $\{z_{\min}, z_{\max}\} = \tilde{p}^{-1}(u)$ so that $\tilde{p}(z_{\min}) = u$ and $\tilde{p}(z_{\max}) = u$. However, usually the inverse transformation $\tilde{p}^{-1}$ is not available. The standard procedure is to assume some interval and then adapt it. We can start with some initial interval and check the end points $z_{\min}$ and $z_{\max}$. We can then increase or decrease the interval end points based on whether $\tilde{p}(z_{\min})$ is larger or smaller than u. The red horizontal line in figure 2 depicts the adapted sampling interval. After adaptation, the next sample is sampled uniformly from the interval.

The slice sampling procedure ensures that the sample from the higher density of the probability distribution is always accepted with probability 1 and sample from lower density is accepted with probability u.

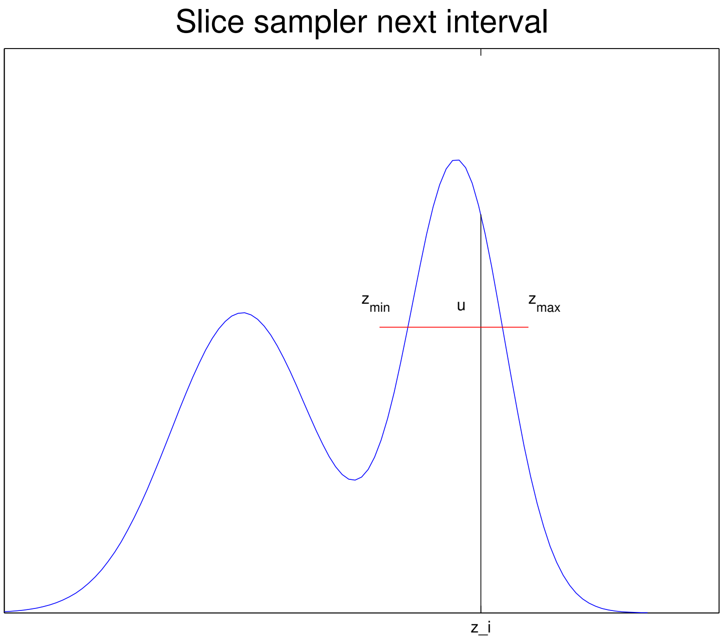


Figure 2: The slice sampler procedure. The current slice, denoted as the black vertical line, is sampled for the variable u which defines the interval of the next sample. This interval, denoted as the red horizontal line, is then adapted so that when we sample from it uniformly, we obtain a sample from the interval defined by the intersections of the distribution and the red line.

5.2.2 Elliptical Slice sampling

Elliptical slice sampling (ESLS, Murray et al., 2009) is a simple parameter-free sampling technique based on the original slice sampling. ESLS can be used when the joint prior distribution of the sampled variables $\mathbf{f}$ is multivariate Gaussian and the posterior can be expressed as

$$p(\mathbf{f} \mid \text{Data}) \propto p(\text{Data} \mid \mathbf{f})p(\mathbf{f}) = p(\text{Data} \mid \mathbf{f})\text{N}(\mathbf{f} \mid \mathbf{0}, \Sigma). \tag{190}$$

The posterior distribution of the latent variables for Gaussian processes is equal to (190) if we replace $\{\text{Data}\}$ with $\{\mathbf{y}, X\}$. Thus, ESLS can be used for sampling the latent variables in (187).

ESLS builds upon the Metropolis-Hastings algorithm, where the new sample for $\mathbf{f}$ can be expressed as

$$\mathbf{f}' = \sqrt{1-\epsilon^2}\mathbf{f} + \epsilon\boldsymbol{\nu}, \qquad \boldsymbol{\nu} \sim \text{N}(\mathbf{0}, \Sigma), \tag{191}$$

with the acceptance probability of

$$p(\text{accept} \mid \mathbf{f}') = \min(1, p(\text{Data} \mid \mathbf{f}')/p(\text{Data} \mid \mathbf{f})). \tag{192}$$

The new state $\mathbf{f}'$ can alternatively be expressed as

$$\mathbf{f}' = \boldsymbol{\nu}\sin\theta + \mathbf{f}\cos\theta, \tag{193}$$

which defines an ellipse passing through $\mathbf{f}$ and $\boldsymbol{\nu}$. ESLS works by first sampling the proposal step $\boldsymbol{\nu}$ from the prior distribution, and then using the slice sampler to find appropriate θ from the ellipse. The ESLS procedure for getting a new sample $\mathbf{f}'$ can be summarized as

1. Draw random number u unifromly from the interval $[0, 1]$ and sample $\boldsymbol{\nu}$ from $\mathrm{N}(\mathbf{0}, \Sigma)$.
2. Sample initial θ unifromly from $[0, 2\pi]$. Set the initial interval of the slice sampler as $\theta_{\mathrm{min}} = \theta - 2\pi$ and $\theta_{\mathrm{max}} = \theta$. Update $\mathbf{f}' = \mathbf{f}\cos\theta + \boldsymbol{\nu}\sin\theta$.
3. If $u \geq p(\text{accept} \mid \mathbf{f}')$, accept the current sample $\mathbf{f}'$.
4. If the current sample is not accepted, shrink the slice sampler interval (the angle of the ellipse) by setting $\theta_{\mathrm{min}} = \theta$ if $\theta < 0$ and $\theta_{\mathrm{max}} = \theta$ otherwise.
5. Sample θ uniformly from $[\theta_{\mathrm{min}}, \theta_{\mathrm{max}}]$. Compute the new proposal $\mathbf{f}'$ and compute the new acceptance probability. If $u \geq p(\text{data} \mid \mathbf{f}')$, accept the sample, otherwise go to 4.

6 Experiments

Our aim in this thesis is to develop and test how the algorithm for Big Data with monotonicity constraint compare to the EP monotonicity constraint. To be able to compare the models we use small and large data sets, some simulated and some real world data sets. We compare 4 different monotonic models in this thesis: Monotonic Gaussian process with Stochastic Variational Inference (**SVI**); Monotonic Gaussian process with Expectation Propagation (**EP**); Monotonic Gaussian process with Markov Chain Monte Carlo sampled parameters and EP approximation to the marginal likelihood (**MCMC**); and Monotonic Gaussian process with a Markov Chain Monte Carlo sampling of both the parameters and the latent values (**FMCMC**). We also compute the results with the non-monotonic standard full GP model (**GP**) and the FIC sparse approximation (**FIC**). EP approximation is used with the full GP and FIC for the classification data sets.

The $O(n^3)$ computational complexity of the standard GP methods restrict the use of these methods to a few thousand data points. The SVI can theoretically handle very large data sets, due to the $O(nm^3)$ complexity. The FIC sparse method can also handle large data sets with the computational complexity of $O(mn^2)$. For data sets larger than 3000 observations, we sample the training data repeatedly for a subset of 1000 observations. These subsets of the original datasets are used to compute the results for the test data set. This kind of subsampling is necessary for other methods besides SVI.

Based on the earlier work on GP models, we expect that EP will be better than SVI (Nickisch and Rasmussen, 2008) if we use the same data for both algorithms. Our interest is in whether the SVI increases the predictive accuracy with larger data sets, when compared to EP results with the subsets of the data. We also compare the monotonic methods (SVI, EP, MCMC, FMCMC) to non-monotonic methods (GP, FIC) to see whether the monotonicity helps or not.

The number of inducing variables for SVI and FIC (the same inducing variables are used for both methods) is 100 for *synthc1* and *synthr* and 1000 for the rest of the data sets. The locations of the inducing variables are chosen with K-means algorithm from the training data.

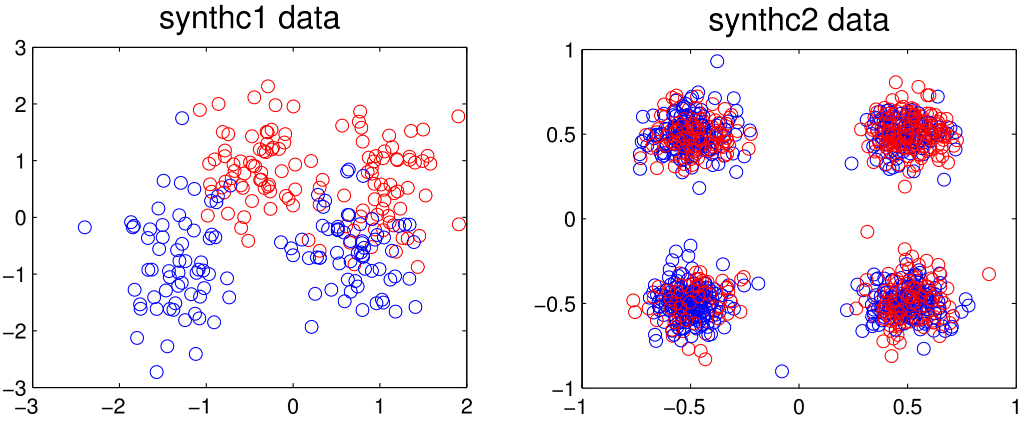
The experiments are done with the modified version of the GPstuff toolbox (Vanhatalo et al., 2013) for `MATLAB` and Octave.

6.1 Data sets

We use several different data sets to assess how SVI algorithm compares to EP. We chose the the real data sets based on the number of observations and they are mainly for comparing SVI and EP. We note that the monotonicity assumptions on these data sets may or may not help with respect to the predictive performance.

6.1.1 Simulated data sets

The models are compared with three simulated data sets. Two of the data sets are for binary classification and one for a regression task. The first binary classification



(a) Synthc1

(b) Synthc2

Figure 3: Example of the simulated classification data sets. The red circles represent class 1 and the blue circles represent class -1.

data set, *synthc1*, is represented in figure 3a. This small data set is mainly for comparing EP and SVI algorithms. The data set consists of 250 observations with 125 for class 1 and 125 for class -1 (Ripley, 1996).

The second simulated binary classification data, *synthc2*, is presented in figure 3b, where we have plotted every 10th data point. The whole data set consists of 10000 training observations and 900 test observations. The training data inputs were sampled from each of the four normal distributions seen in figure 3b, with means defined by $[0.5, 0.5]^T$, $[0.5, -0.5]^T$, $[-0.5, 0.5]^T$ and $[-0.5, -0.5]^T$. The covariance for each normal distribution was diagonal with the variance of 0.1^2 for both dimensions. Each normal distribution was sampled for 2250 points. The true latent function was linear with

$$f(\mathbf{x}_i) = 0.5x_{i,1} + 0.5x_{i,2}. \tag{194}$$

The observations y_i were created by adding Gaussian noise $\epsilon_i \sim \mathrm{N}(0, 1)$ to the true latent function and then thresholding the latent function with

$$y(\mathbf{x}_i) = \begin{cases} 1 & \text{if } f(\mathbf{x}_i) + \epsilon_i \geq 0, \\ -1 & \text{if } f(\mathbf{x}_i) + \epsilon_i < 0. \end{cases} \tag{195}$$

The test data set was created by taking the test inputs from a uniform grid of 30×30 in the interval $[-1, 1] \times [-1, 1]$ and then forming the test latent function and observations without adding noise.

The simulated regression data, *synthr*, set is presented in figure 4. This training data was created by first sampling 225 data points uniformly from $[-2, 2] \times [-2, 2]$ and then creating the latent function

$$f(\mathbf{x}_i) = 3\Phi(2x_{i,2}) + 2\Phi(4x_{i,1}) + 0.5x_{i,1} + 0.5x_{i,2}, \tag{196}$$

where Φ is the standard cumulative Gaussian function. The observations $y(\mathbf{x}_i)$ were created by adding Gaussian noise $\epsilon_i \sim \mathrm{N}(0, 0.25^2)$ to the latent function values. The

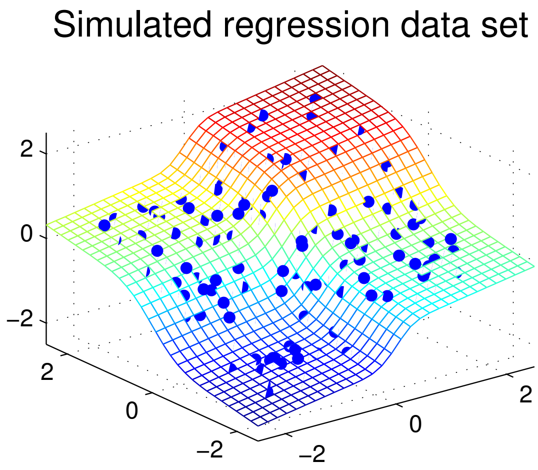


Figure 4: Simulated regression data set *synthr*. The blue dots are the noisy observations and the surface represents the true function.

test data was created by taking 900 test inputs uniformly from the grid $[-2.5, 2.5] \times [-2.5, 2.5]$ and then forming the test function values without adding noise. This data set is used to demonstrate how the monotonicity can help in extrapolation tasks.

We assume monotonically increasing latent function with respect to both input dimensions for all three simulated data sets.

6.1.2 Arsenic

The arsenic data set consists of 3020 observations with four explanatory variables for each observation and binary response variable. The data set is collected from villagers in Africa. The explanatory variables are 1. Arsenic content of the water, 2. distance to the next well, 3. participation of the villagers in the village association and 4. how much the villagers have been educated about the dangers of arsenic. The binary response variable is 1 if the villagers switch to use other well and 0 if they continue to use the same well. We assume that the latent function is monotonically increasing with respect to the first input variable, meaning that the more there is arsenic in the water, the more likely the villagers are to change to the next well. We also assume that the response variable decreases monotonically with respect to the second input variable, meaning that the longer the distance to the next well, the less likely the villagers change to use that well.

6.1.3 Leukemia

The leukemia data set consists of 1043 survival times t_i and censoring indicators for people with acute myeloid leukemia, between the years 1982 and 1998 from the United Kingdom Leukemia Register. The censoring indicator tells whether the event (death) has happened or not at the recorded survival time. The explanatory variables are age, sex, white blood cell count at the time of leukemia diagnosis, and Townsend deprivation index which measures the deprivation in the district of residence.

The data set can also be considered with respect to the *person-moments*, where the 'moment' is a point in time for each person (Hanley and Miettinen, 2009). The observations are then the last moments where either the event happens (not censored) or does not happen (censored). We want to model the hazard, or the likelihood, of the event happening for person x at time t. T he person-moments effectively transform the data set to infinite size: 1043 time points with the event happening (not censored) or not happening (censored) and an infinite number of time points before the observation time t_i when the event has not happened (continuous time).

To be able to do inference for the person-moment data, we sample a subset of the time points for the event not happening (Mantel, 1973; Hanley and Miettinen, 2009). Thus, the data set consists of the original observation triplets x_i, t_i, y_i where we have the time of the event t_i, an indicator of whether the event happened $y = 1$ or was cencored $y = -1$, and the explanatory variables for each person x_i. In addition to these, the data set consists of the sampled triplets $x_i, t_{i,s}, y_{i,s}$ where we sample the times $t_{i,s}$ for each person x_i with $y_{i,s} = -1$ always (event not happened).

The times $t_{i,s}$ are sampled so that we have a total of 10000 observations, meaning that we sample $1000 - 1043 = 8953$ time points in total. The number of time points sampled for person x_i is proportional to the duration to the event for that person t_i. This means that if we have for example person x_1 and person x_2 with event times $t_1 = 100$ and $t_2 = 50$, we expect there to be twice as many samples $t_{1,s}$ than $t_{2,s}$.

The final analysis is done using the binary event indicators y_i and $y_{i,s}$ with explanatory variables x_i and t_i. The task is then to predict whether the event happens or not for some pair $\{x, t\}$. We assume monotonic increase for the likelihood of the event with respect to the age in x_i.

6.1.4 Adult

Adult data set (Kohavi, 1996) consists of 30725 observations for the training data set and 15318 observations for the test data set. The data set is extracted from the census bureau database found at http://www.census.gov/ftp/pub/DES/www/welcome.html.

The binary response variable y in the data set is whether a persons income exceeds 50 000$ a year. The explanatory variables are age, work class, education, marital status (numeric value between 1-7 depending of whether the person is for example single or widowed), relationship status (1 if the person is married and 0 otherwise), race (1 if the person is white, 2 if black and 3 otherwise), sex, and working hours per week.

We assume a monotonically increasing latent function with respect to the first and third explanatory variables, meaning that we assume that the older and higher educated the person is, the more likely he or she has an income of over 50 000 $ a year.

6.2 Comparing the models

The models are compared by computing the mean log predictive densities (MLPD) of the predictions for the independent test data set

$$\mathrm{MLPD}_M = \frac{1}{n_t} \sum_{i=1}^{n_t} p(y_i^* \mid \mathbf{x}_i^*, X, \mathbf{y}, M), \tag{197}$$

where y_i is the true function value at x_i^*, not included in $\mathbf{y}$, and $p(y_i^* \mid \mathbf{x}_i^*, X, \mathbf{y}, M)$ is the predictive distribution of the y_i^*, evaluated at y_i^*, for model M. Independent test data set means that we don't use it for training the model. For regression tasks, the predictive distribution is Gaussian with predictive mean and variance, and log predictive density is just the logarithm of the Gaussian distribution evaluated at y_i^*. For classification, the predictive density is the predicted probability of the corresponding class and log predictive density is the logarithm of the probability.

If the independent test set $\{X^*, \mathbf{y}^*\}$ is not available, we can use cross-validation to approximate the predictive performance, using only the training data $\{X, \mathbf{y}\}$. For the binary classification data sets we also use the receiver operating characteristics (ROC) curve to assess the predictions.

As noted earlier, the larger data sets (*synthc2*, *leukemia*, *adult*, *bike*) are sampled randomly to get a the subset of 1000 observations from the original data set. This sampling is necessary, in order compute the predictions for the computationally costly methods (EP, MCMC, FMCMC, GP). The subsampling is repeated several times and the predictions for the test data set are averaged over the different training subsets. The subsampling allows the measurement whether the SVI algorithm learns additional information from the whole data set, compared to the subset of the data sets. The smaller data sets (*synthc1*, *synthr*, *arsenic*) are used to measure how well the SVI algorithm performs against EP and MCMC methods.

6.2.1 Cross-validation

Bayesian cross-validation (Vehtari and Lampinen, 2002; Vehtari and Ojanen, 2012) is a method for assessing the predictive quality of statistical methods. Bayesian cross-validation can be used to estimate how the model would perform on an independent test data set.

In cross-validation the training data set $\{X, \mathbf{y}\}$ is split into k parts, indexed by $\{I_1, I_2, \ldots, I_k\}$. The model is then trained without one of the subsets I_i. The model predictions are computed for input points x_j, $j \in I_i$ and the predictions are tested against y_j. This is repeated k times, once for every i, to get the approximate test predictions for the training data set. The MLPD can be approximated with

$$\mathrm{MLPD}_M \approx \sum_{i=1}^{k} \frac{1}{|I_i|} \sum_{j \in I_i} p(y_j \mid \mathbf{x}_j, X_{-I_i}, \mathbf{y}_{-I_i}, M) = \frac{1}{n} \sum_{j \in I_i} p(y_j \mid \mathbf{x}_j, X_{-I_i}, \mathbf{y}_{-I_i}, M). \tag{198}$$

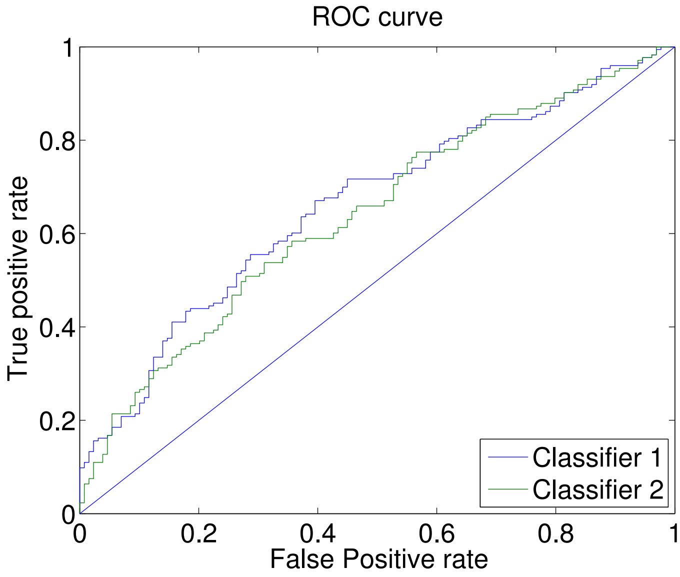


Figure 5: An example of a ROC curve for two different classifiers. Here classifier 1 is the better classifier as we expect more true positives and less false negatives than with classifier 2.

6.2.2 Reicever operating charachteristics

Receiver operating characteristics (ROC curve) visualizes the performance of a binary classifier. ROC curve plots the true positive rate (TPR), or *sensitivity*, against the false positive rate (FPR), or *fall-out*, of the classifier, by varying the threshold value of classification to one of the classes. The true positive rate is the number of correctly classified test instances for class 1 divided by the true number of test instances belonging to class 1. The false positive rate is the number of falsely classified test instances belonging to class 0 divided by the true number of test instances belonging to class 0.

For GP models, the ROC curve is computed by varying the probability threshold tr of predicting to class 1

$$y^* = \begin{cases} 1, \text{if } p(y^* = 1 \mid \mathbf{x}^*, X, \mathbf{y}) > tr, \\ 0, \text{otherwise}. \end{cases} \tag{199}$$

An example of a ROC curve is in figure 5, where the different colors represent different methods. The diagonal line in a ROC curve is the pure guess, meaning that we expect as many false positives as true positives. The more the ROC curve bends to the upper left corner, the better the classifier, because we expect less false positives while getting more true positives.

Data set	SVI	EP	MCMC	FMCMC
synthc1	-0.339	-0.339	-0.343	-0.349
synthc2	-0.470 ± 0.0067	-0.457 ± 0.0138	-0.454 ± 0.0191	-0.385 ± 0.0058
synthr	0.722 ± 0.0945	0.906 ± 0.1077	0.918 ± 0.0680	-1.44 ± 4.45
Arsenic	-0.647	-0.640	-0.641	-0.640
Leukemia	-0.255	-0.261	-0.260	-0.263
Adult	-0.365	-0.375 ± 0.0023	-0.374 ± 0.0022	-0.386 ± 0.0072

(a) Monotonic methods

Data set	GP	FIC
synthc1	-0.305	-0.305
synthc2	-0.493 ± 0.0087	-0.475 ± 0.0092
synthr	0.666 ± 0.0956	0.661 ± 0.0959
Arsenic	-0.640	-0.640
Leukemia	-0.258	-0.249
Adult	-0.376 ± 0.0042	-0.358

(b) Non-monotonic method

Table 1: The mean log predictive values for the different methods. For the cases where we have different realizations of the training data, the uncertainty of the MLPD values is shown as the mean over the repetations ± one standard deviation. These cases include the cases where we sample a subset of the data as well as the simulated data sets. Note that even though we sample a subset of data for *leukemia* data, the uncertainty intervals are not shown, because the MLPD values are computed with cross-validation.

7 Results

Table 1a displays the mean log predictive densities for the monotonic methods. Table 1b displays the results for standard GP methods. In the table 1b, the GP refers to either standard GP regression solution (regression data) or the EP approximation (classification data).

The results of the small data sets (*synthc1*, *synthr*, *arsenic*) confirm that the SVI approximation to the marginal likelihood is not as good as the EP approximation with respect to predictive density. Figure 6 displays the ROC curve for the *synthc1* data set. The ROC curve confirms that both the EP and SVI algorithm work for this data set and there are not much differences with respect to predictions. Figure 7 displays the predictions of the latent function $\mathbf{f}$ for different methods. We see that both the EP and SVI produce identical function surfaces which differ somewhat from the non-monotonic methods.

The results on the second simulated classification data set, *synthc2*, are worse with SVI because the data set is quite simple and the additional data points don't help in the prediction task. Figure 8 displays the ROC curve for the *synthc2* data set. Figure 9 displays the predictions of the latent function.

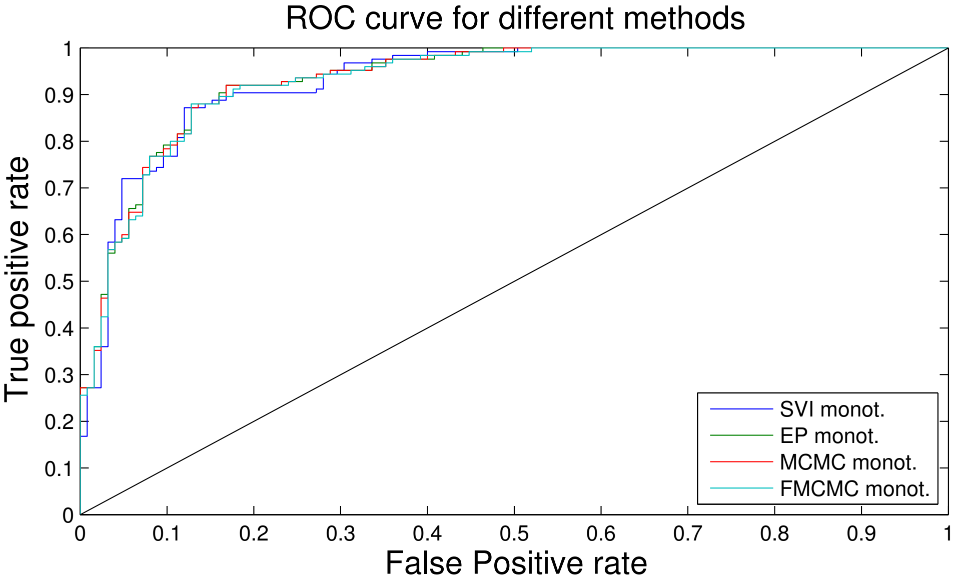


Figure 6: ROC curve and the training data for the *synthc1* data.

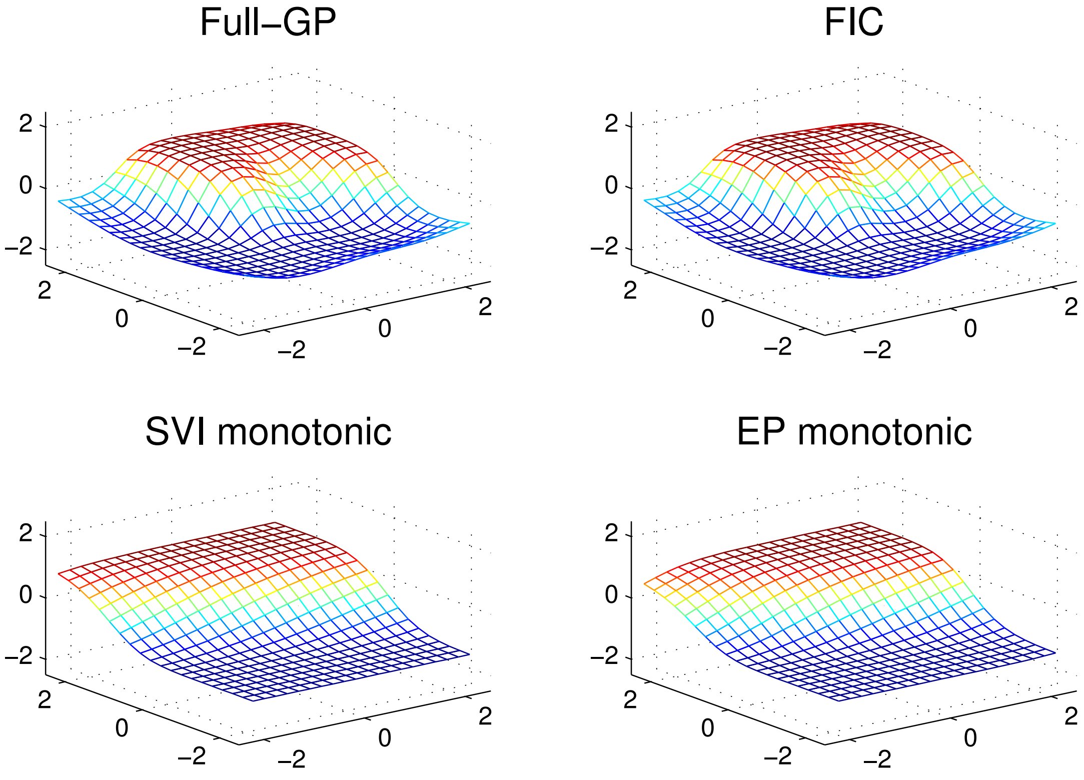


Figure 7: Latent function for different methods for the *synthc1* data.

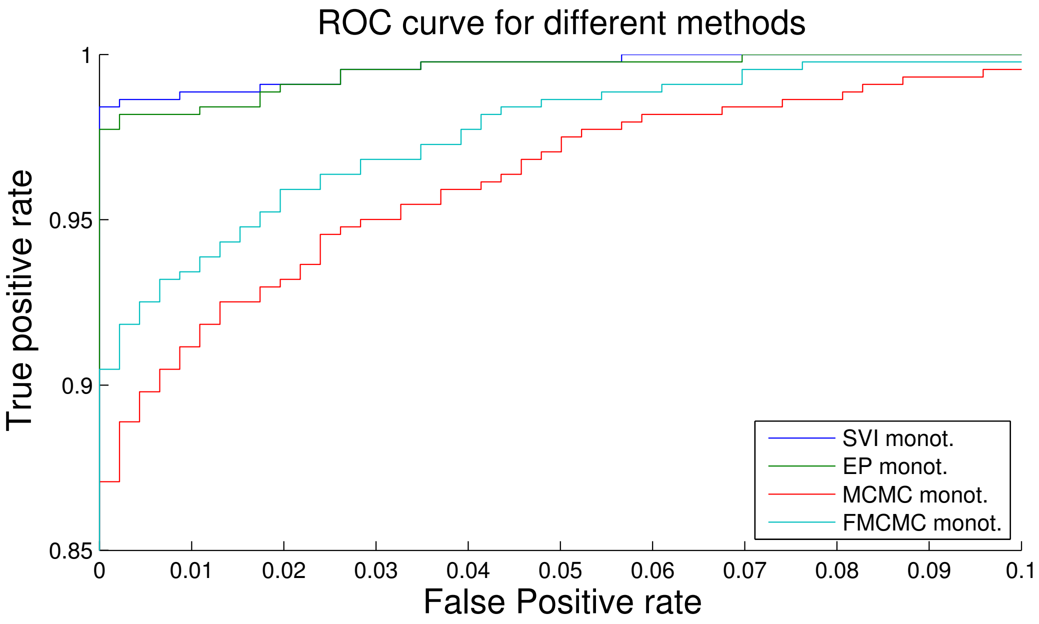


Figure 8: ROC curve and the training data for the *synthc2* data. Note that every methods works very well for this data and thus we have limited the ROC curve for the upper left quadrant of the normal $[0, 1] \times [0, 1]$ limits.

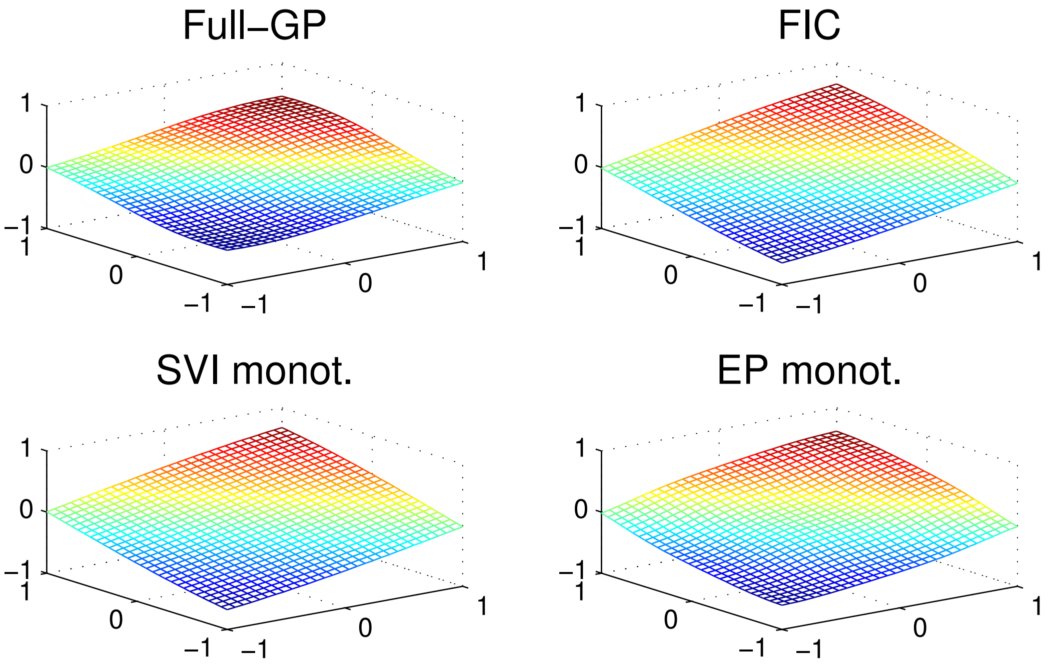


Figure 9: Latent function for different methods for the *synthc2* data.

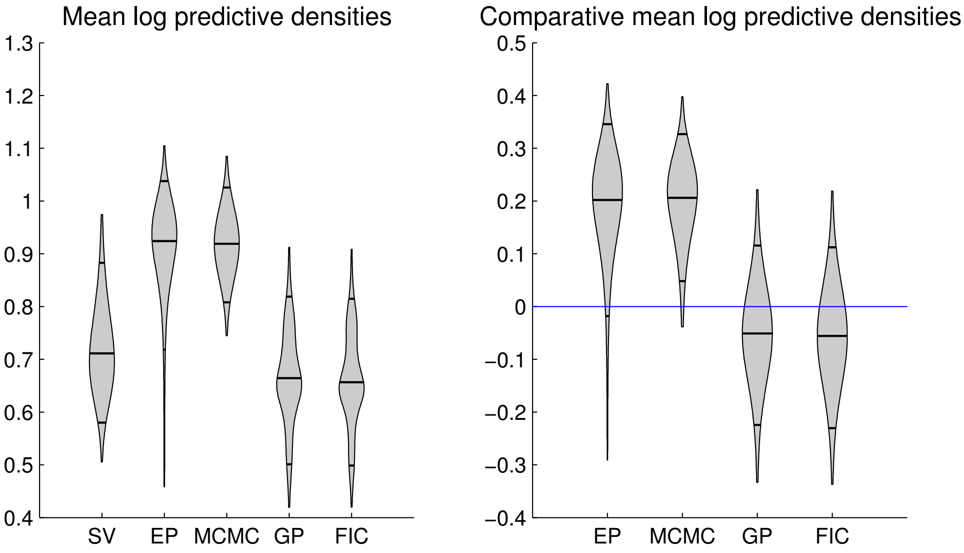


Figure 10: Density plots of the MLPD values for *synthr* data set. GP denotes the full non-monotonic GP. The right density plot is the densities of the methods compared to the SVI predictions. The proportion below zero has MLPD lower than with SVI and above zero higher. The density plots are formed over 100 simulations of the training data.

Figure 10 plots the density of the mean log predictive densities over 100 simulations of the training data for the *synthr* regression task. We see that the EP algorithm works better here than the SVI. For this data set, the test data set includes observations which are outside the convex hull of the training data set, so predicting becomes an extrapolation task. The monotonicity constraint takes away some of the flexibility of the function and the predictions are better. This is due to the additional information monotonicity provides. Extrapolation with Gaussian processes is generally very hazardous as the uncertainties increase rapidly the further the test inputs are from the training inputs. Monotonicity constraints the latent function on this extrapolation task and thus the predictions are better. This monotonicity constraint can be seen in the figure 11, where the predictions of the latent function are displayed.

ROC curve for the *arsenic* data set in figure 12 confirms the fact that EP works better than SVI if we use the same data. While there is not a great difference, it is obvious that the EP approximation is better than the SVI approximation. Figure 13 displays the conditional predictions for the different explanatory variables of the *arsenic* data. The conditional predictions are done by fixing the other explanatory variables to their median value from the training data set and then varying the specific explanatory variable. We see that the conditional latent predictions are almost the same. It is obvious that the SVI approximation prefers smoother functions in this case. This can be due to the number of inducing variables or their placement.

The monotonicity does not help on the *synthc1* or arsenic data, but the results

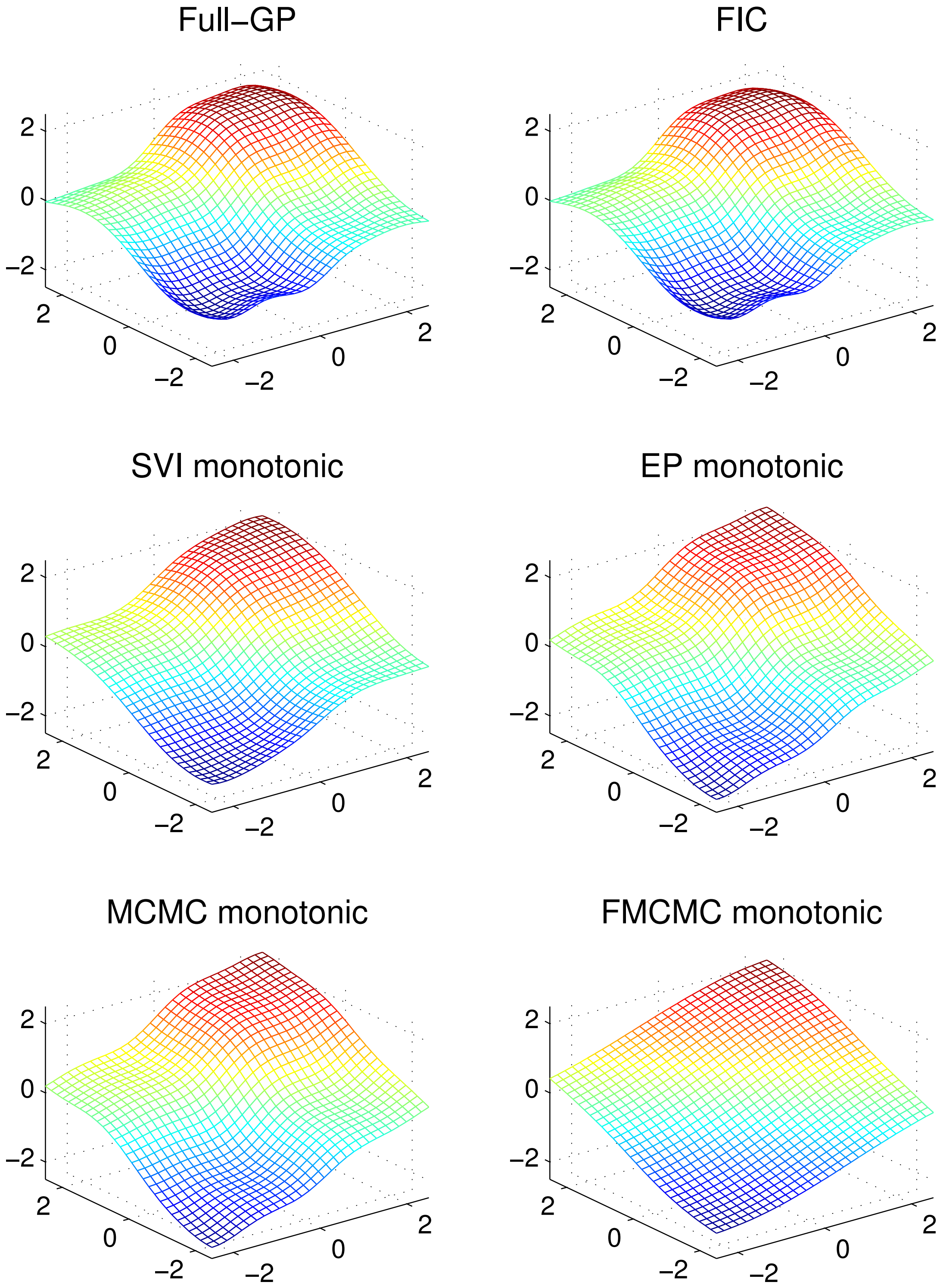


Figure 11: Latent function predictions for the *synthr* data.

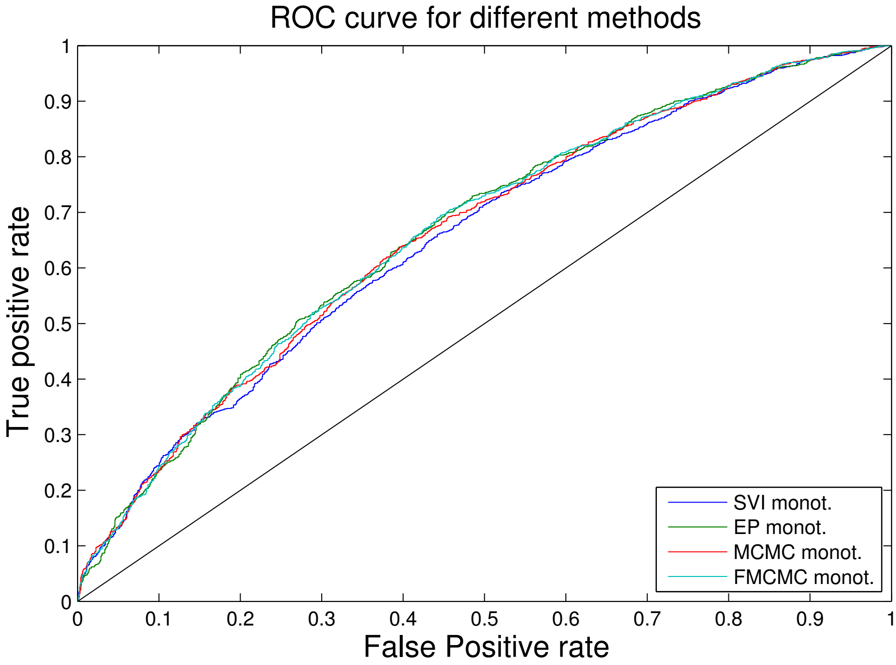


Figure 12: ROC curve for the *arsenic* data set.

on *synthr* are better with the assumed monotonicity. Both the SVI and EP are competitive with respect to the MCMC results. We see from the results that the full MCMC fails in the simulated regression task. This is due to the elliptical slice sampler getting stuck on a unfavorable (non-monotonic) solution which causes the samplers to accept all kinds of weird samples.

When using the subset of the whole data for EP and MCMC methods with the larger data sets (*leukemia*, *adult*), we see that the results are better with SVI. We can thus argue that the SVI monotonicity learns additional information from the data set and gives better predictions.

The ROC curve for the leukemia data is displayed in figure 14. The conditional predictions are in figure 15. We see that while there is not much difference, SVI is better than EP and MCMC methods. The conditional predictions also differ for explanatory variable time but for the other explanatory variables, they seem to be almost identical. Again, SVI prefers more linear latent functions.

The ROC curve for the adult data in 16 supports the conculusion that the SVI algorithm learns from the whole data. We see that the SVI performs better than the EP and MCMC methods with subsampled data. The conditional predictions for the *adult* data in figure 17 agree with the earlier conclusions that the SVI and EP produce almost identical conditional predictions.

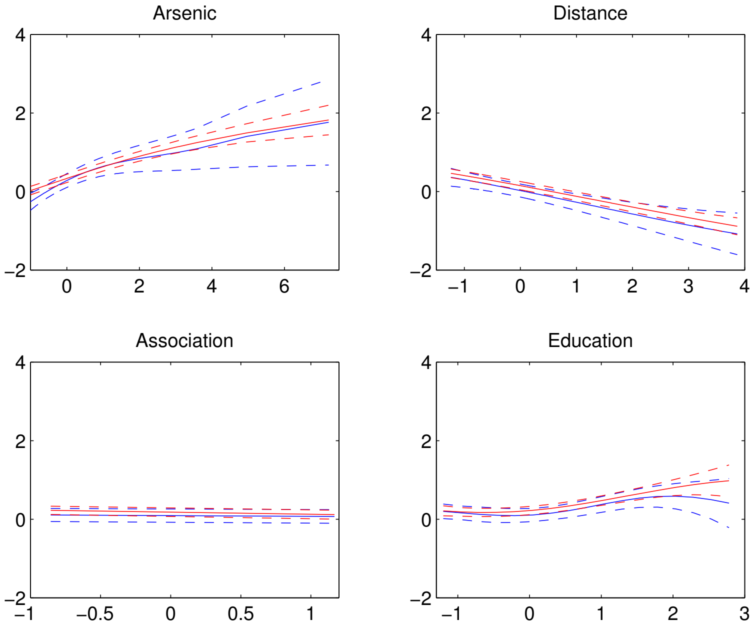


Figure 13: Conditional predictions for the *arsenic* data. The red line is the SVI conditional prediction and the blue line is the EP conditional prediction. Dashed lines represent ± three standard deviations of the predictions. Monotonic increase of the latent function is assumed with respect to arsenic content in the water (upper left) and monotonic decrease with respect to the distance (upper right). The y-axis represents the latent function prediction and on the x-axis are the values of the specific covariate.

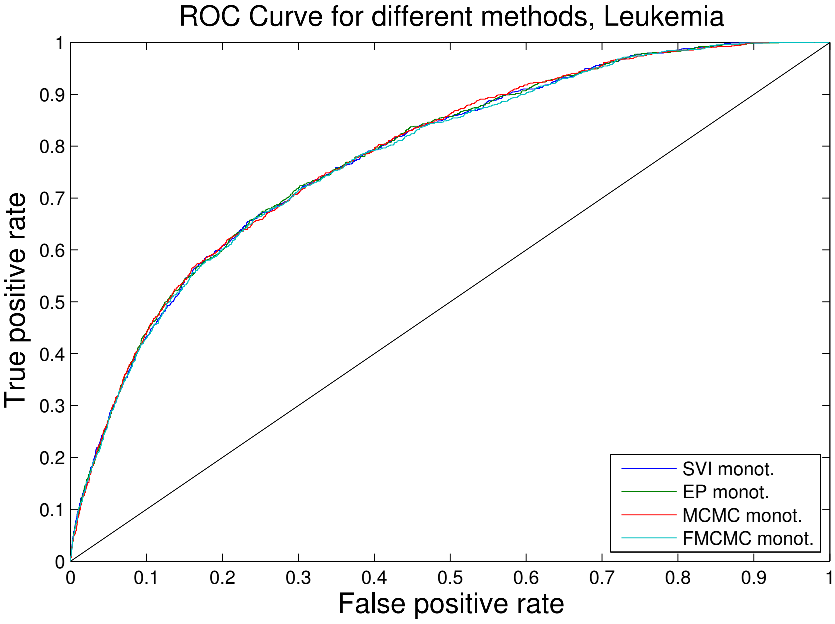


Figure 14: ROC curve for *leukemia* data set.

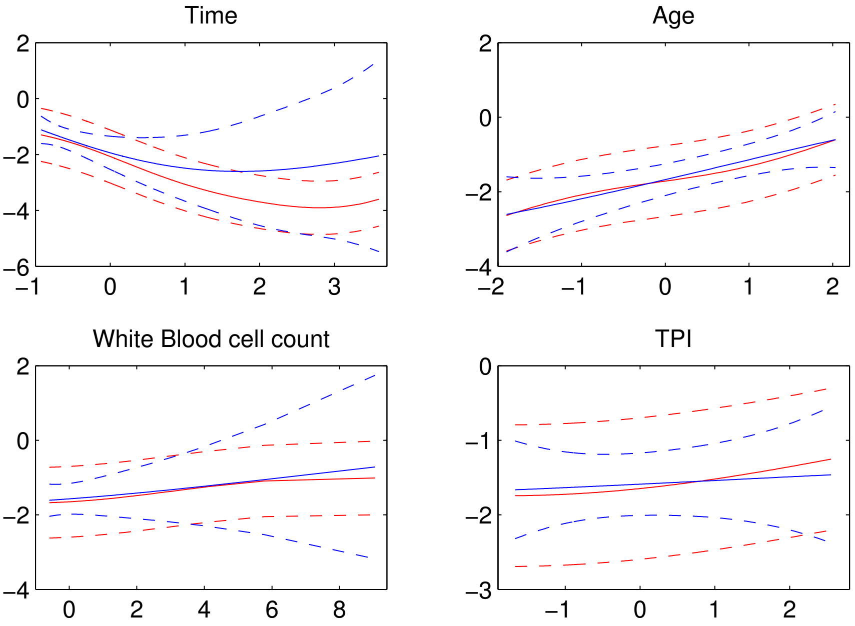


Figure 15: Conditional predictions for *leukemia* data set. The red line is the SVI conditional prediction and the blue line is the EP conditional prediction. Dashed lines represent ± three standard deviations of the predictions. Monotonic increase of the latent function is assumed for the age (upper right)

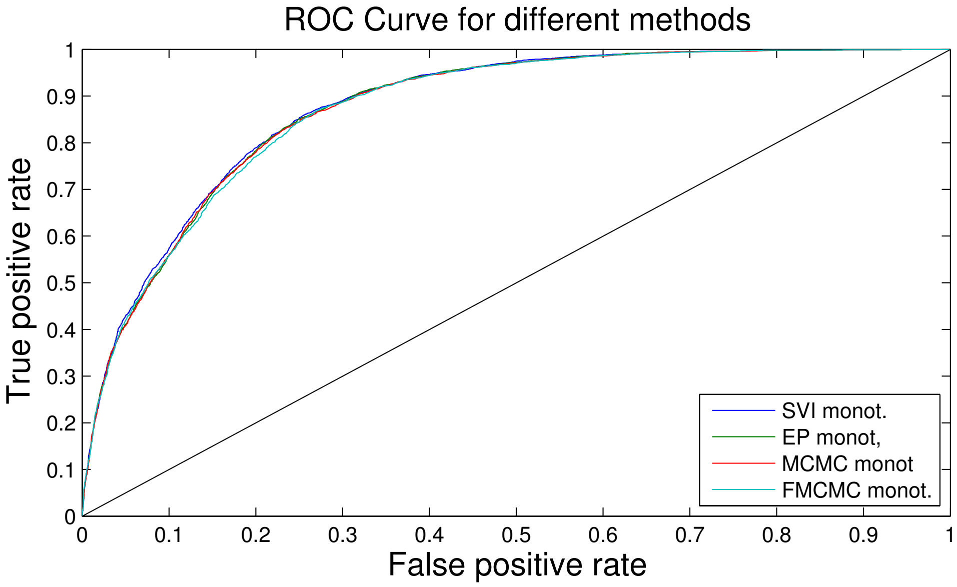


Figure 16: ROC curve for *adult* data set.

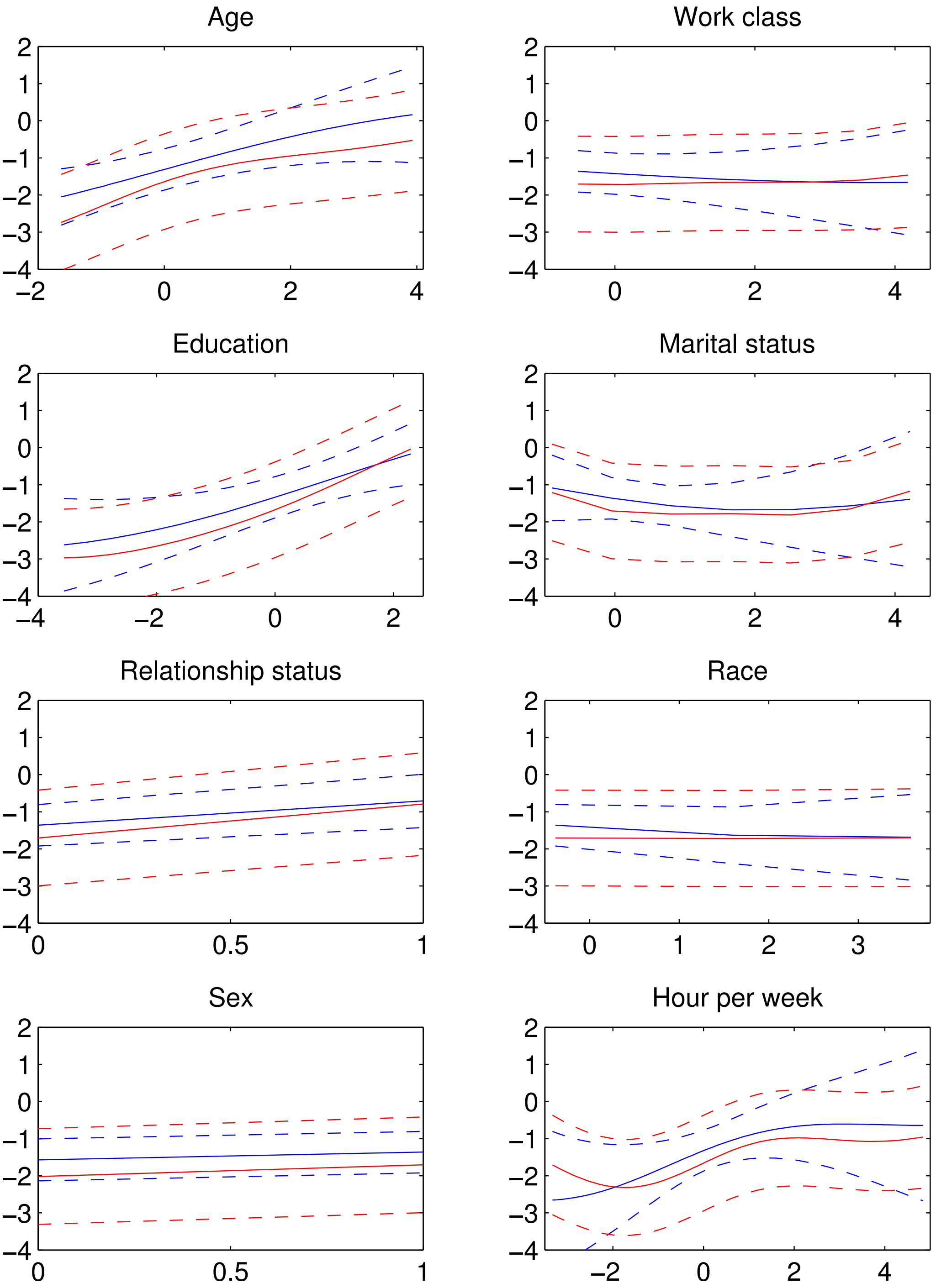


Figure 17: Conditional predictions for *adult* data set. Blue line corresponds to EP and red line for SVI. Monotonic increase of the latent function is assumed for the age (first up, left) and the education (second up, left).

8 Discussion

In this thesis, we developed a method for using monotonicity information for Gaussian processes when the data sets are large. The stochastic variational inference (SVI) based method was experimented with several different data sets, both simulated and real world data. The new method was tested against the Expectation Propagation (EP) based monotonic method, as well as the sampling based methods (MCMC) which can be thought of as the ground truth with the limit of an infinite number of samples. We also tested how the monotonic methods fare against standard non-monotonic methods.

The aim of this thesis was also to show that the SVI can be applied straightforwardly to also infer the virtual monotonicity observations, and that the resulting algorithm can be applied to several different kinds of problems. We wanted to experiment how the SVI algorithm fares against the EP algorithm which is known to produce very good results with binary classifiers.

The experiments showed that the SVI algorithm can be used for large data sets and yields better results than the computationally heavy EP or MCMC methods for the subsets of the data. The hard part in including monotonicity constraints is that if these are not reasonable assumptions, the non-monotonic methods are usually better with respect to the predictions. However, it was shown that, while the monotonicity constraints might restrict the function too much, the SVI algorithm is a viable option when there is indeed knowledge of the monotonicity.

The proposed monotonicity constraint rely heavily on the choice of the virtual inputs. In the ideal situation, we would have an infinite number of virtual inputs all over the input space. However, while the SVI can theoretically handle very large number of observations, the computation becomes quite slow if there is a great number of input dimensions. Usually, we settle for a small number of virtual inputs, which are inside the convex hull of the true observed input values, as long as the forced function is monotonic in the observed inputs and the virtual inputs. In this thesis, we used the K-means algorithm to select the virtual inputs. If there are a small number of dimensions (3 or less), a viable option is to simply set the virtual inputs in a uniform grid. While the use of virtual observations to force monotonicity might seem like an arbitrary and not so elegant choice, it also has some interesting properties that can be of use. The use of virtual observations naturally enables the choice of different monotonicity for different input dimensions. Furthermore, by placing the virtual observations appropriately, one can also induce for example the unimodality of the latent function. This can be achieved by assuming that the latent function is monotonically increasing (or decreasing) for the values of the input which are smaller than the assumed mode of the latent function and monotonically decreasing (increasing) for the values of the input which are greater than the mode. This can be readily achieved by setting the virtual observations $y_v = 1$ where we assume the function is increasing and $y_v = -1$ for where it is assumed to be decreasing.

The use of the binary classifier for the virtual observations causes the function to prefer larger gradients. This can cause problems in regions where the function might be flat. This has already been discussed in the earlier work by Riihimäki and Vehtari (2010). The virtual inputs can be used again to allow flat or semi-monotonic function by not setting any virtual inputs in that region. Another possible way is a multi-class classifier, where we could have the virtual observations of -1, 0 and 1, based on whether the function is monotonically decreasing, flat, or monotonically increasing. The likelihood for the flat observations could for example be standard Gaussian so that the latent function values are penalized based on how far they are from zero.

The use of latent likelihood for the binary observations with SVI can also cause some problems. Optimizing the noise variance parameters for the latent Gaussian likelihood can easily cause the likelihood to overfit. On the other hand, it can allow missclassifications too easily. For this reason, we chose to not optimize the noise variance for the virtual observations as we want the likelihood to penalize latent values which don't agree with the monotonicity assumption.

Because the stochastic nature of the SVI algorithm, care also needs to be taken for the updating schedule. We followed the suggestions of Hensman et al. (2013) and fixed the hyperparameters for a few epochs of the training data to allow the variational parameters to update to some reasonable values. Based on our experiments the stochastic nature can very easily cause problems with the updates of the variational parameters and the step-size parameter should thus be very conservative. This may result in the need of few more iterations, but we accept this. Sometimes it is also necessary to force the positive definitiness of the variational covariance S, which can easily be done with eigenvalue decomposition.

We already mentioned in section 7 that the samplers can occasionally get stuck. This was emphasized in the *synthr* experiment. We noted that sometimes the ESLS accepted latent values which were not monotonic. Accepting these samples caused the marginal likelihood to decrease to very small values and the subsequent samples to be accepted whether they were reasonable or not. One possible approach to eliminate this would be to sample the latent values and hyperparameters and accept only the pairs where the latent function was monotonic. However, pruning like this can cause all of the samples to be rejected after the sampler gets stuck.

This thesis introduced the SVI application to the monotonicity constraints with virtual observations. The experiments provided valuable information which strengthened the prior assumption that EP approximation works better than the variational approximation. The experiments also showed that the SVI application can be applied to large data sets, and it utilizes the whole data set in learning. The prior experimental questions were thus answered and we can consider the work to be successful.

References

Amari, S.-I. (1982). Differential Geometry of Curved Exponential Families-Curvatures and Information Loss. *The Annals of Statistics*, pages 357–385.

Amari, S.-I. (1998). Natural Gradient Works Efficiently in Learning. *Neural computation*, 10(2):251–276.

Bayes, T. (1763). A Letter From the Late Reverend Mr. Thomas Bayes, frs to John Canton, ma and frs. *Philosophical Transactions*, 53:269–271.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*, volume 4. Springer New York.

Boyd, S. P. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge university press.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. CRC press.

Hanley, J. A. and Miettinen, O. S. (2009). Fitting Smooth-In-Time Prognostic Risk Functions via Logistic Regression. *The International Journal of Biostatistics*, 5(1).

Hastings, W. K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian Processes for Big Data. *arXiv preprint arXiv:1309.6835*.

Hensman, J., Rattray, M., and Lawrence, N. D. (2012). Fast Variational Inference in the Conjugate Exponential Family. *Advances in neural information processing systems*, pages 2897–2905.

Hestenes, M. R. and Stiefel, E. (1952). *Methods of Conjugate Gradients for Solving Linear Systems*, volume 49. NBS.

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

Kohavi, R. (1996). Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In *Knowledge Discovery and Data Mining*, pages 202–207.

Kullback, S. and Leibler, R. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Mantel, N. (1973). Synthetic Retrospective Studies and Related Topics. *Biometrics*, pages 479–486.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The journal of chemical physics*, 21(6):1087–1092.

Metropolis, N. and Ulam, S. (1949). The Monte Carlo Method. *Journal of the American statistical association*, 44(247):335–341.

Minka, T. P. (2001a). Expectation Propagation for Approximate Bayesian Inference. *Uncertainty in Artificial Intelligence*, 17:362–369.

Minka, T. P. (2001b). *A Family of Algorithms for Approximate Bayesian Inference.* PhD thesis, Massachusetts Institute of Technology.

Murray, I., Adams, R. P., and MacKay, D. J. (2009). Elliptical Slice Sampling. *arXiv preprint arXiv:1001.0175.*

Neal, R. (1997). Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification. *Arxiv preprint physics/9701026.*

Neal, R. M. (2003). Slice sampling. *Annals of statistics*, pages 705–741.

Nickisch, H. and Rasmussen, C. E. (2008). Approximations for Binary Gaussian Process Classification. *Journal of Machine Learning Research*, 9:2035–2078.

Opper, M. and Winther, O. (2000). Gaussian processes for classification: Mean-field algorithms. *Neural Computation*, 12(11):2655–2684.

Parisi, G. (1998). *Statistical field theory*, volume 28. Perseus Books New York.

Quinonero-Candela, J. and Rasmussen, C. (2005). A Unifying View of Sparse Approximate Gaussian Process Regression. *The Journal of Machine Learning Research*, 6:1939–1959.

Rasmussen, C. E. (2003). Gaussian Processes to Speed up Hybrid Monte Carlo for Expensive Bayesian Integrals. In *Bayesian Statistics 7: Proceedings of the 7th Valencia International Meeting*, pages 651–659.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian Processes for Machine Learning.* MIT Press.

Riihimäki, J., Jylänki, P., and Vehtari, A. (2013). Nested Expectation propagation for Gaussian Process Classification with a Multinomial Probit Likelihood. *Journal of Machine Learning Research*, 14:75–109.

Riihimäki, J. and Vehtari, A. (2010). Gaussian Processes with Monotonicity Information. In *International Conference on Artificial Intelligence and Statistics*, pages 645–652.

Ripley, B. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge.

Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The annals of mathematical statistics*, pages 400–407.

Seeger, M. (2005). Expectation Propagation for Exponential Families. Technical report, University of California at Berkeley, 2005.

Sill, J. and Abu-Mostafa, Y. S. (1997). Monotonicity Hints. *Advances in neural information processing systems*, pages 634–640.

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian Processes using pseudo-inputs. *Advances in neural information processing systems*, 18:1257.

Solak, E., Murray-Smith, R., Leithead, W., D.J., L., and Rasmussen, C. (2003). Derivative observations in Gaussian Process models of dynamic systems. *Advances in neural information processing systems*, 15:1033—1040.

Vanhatalo, J., Riihimäki, J., Hartikainen, J., Jylänki, P., Tolvanen, V., and Vehtari, A. (2013). GPstuff: Bayesian Modeling with Gaussian Processes. *Journal of Machine Learning Research*, 14:1175–1179.

Vehtari, A. and Lampinen, J. (2002). Bayesian model assessment and comparison using cross-validation predictive densities. *Neural Computation*, 14(10):2439–2468.

Vehtari, A. and Ojanen, J. (2012). A survey of Bayesian predictive methods for model assessment, selection and comparison *Statistics Surveys*, 6:142–228.

A Derivatives of the covariance function

For the Gaussian processes, the covariance between function value f_1 and derivative f_2' at some other point, can be computed by derivating the covariance between function values f_1 and f_2. The covariance between two derivatives f_1' and f_2' can be computed analogously by differentiating the covariance between f_1 and f_2 twice.

With the Squared-Exponential covariance function

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{\sum_{i=1}^{d}(x_i - x_i')^2}{2l_i}\right), \tag{A.1}$$

the first derivative of the covariance function is

$$\frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial x_g} = \sigma_f^2 \left(-\frac{1}{l_g}\left(x_g - x_g'\right)\right) \exp\left(-\frac{\sum_{i=1}^{d}(x_i - x_i')^2}{2l_i}\right). \tag{A.2}$$

The second derivative is

$$\frac{\partial^2 k(\mathbf{x}, \mathbf{x}')}{\partial x_g \partial x_p} = \sigma_f^2 \frac{1}{l_g} \left(\delta_{gp} - \frac{1}{l_p}\left(x_g - x_g'\right)\left(x_p - x_p'\right)\right) \exp\left(-\frac{\sum_{i=1}^{d}(x_i - x_i')^2}{2l_i}\right), \tag{A.3}$$

where $delta_{gp} = 1$ if $g = p$ and 0 otherwise.

B Gaussian identities

If we have joint Gaussian distribution of two variables $\mathbf{a}$ and $\mathbf{b}$

$$p(\mathbf{a}, \mathbf{b}) = \mathrm{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{m}_a \\ \mathbf{m}_b \end{bmatrix}, \begin{bmatrix} S_a & S_{ab} \\ S_{ba} & S_b \end{bmatrix} \right), \tag{B.1}$$

the conditional distribution $p(\mathbf{a} \mid \mathbf{b})$ is

$$p(\mathbf{a} \mid \mathbf{b}) = \mathrm{N}\left(\mathbf{a} \mid \mathbf{m}_a + S_{ab} S_b^{-1} (\mathbf{b} - \mathbf{m}_b), S_a - S_{ab} S_b^{-1} S_{ba}, \right) \tag{B.2}$$

and the marginal distribution $p(\mathbf{a})$ is

$$p(\mathbf{a}) = \mathrm{N}(\mathbf{a} \mid \mathbf{m}_a, S_a). \tag{B.3}$$

The product of two Gaussian distributions can be computed with

$$\mathrm{N}(\mathbf{x} \mid \mathbf{m}_1, S_1) \mathrm{N}(\mathbf{x} \mid \mathbf{m}_2, S_2) = Z^{-1} \mathrm{N}(\mathbf{x} \mid \mathbf{m}, S), \tag{B.4}$$

where

$$\mathbf{m} = S \left(S_1^{-1} \mathbf{m}_1 + S_2^{-1} \mathbf{m}_2 \right), \tag{B.5}$$

$$S = \left(S_1^{-1} + S_2^{-1} \right)^{-1}, \tag{B.6}$$

$$Z^{-1} = (2\pi)^{D/2} |S_1 + S_2|^{-1/2} \exp\left(-\frac{1}{2} (\mathbf{m}_1 - \mathbf{m}_2)^T (S_1 + S_2)^{-1} (\mathbf{m}_1 - \mathbf{m}_2) \right), \tag{B.7}$$

and D is the dimension of $\mathbf{x}$.