HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Information and Natural Sciences

Jaakko Luttinen

Gaussian-process factor analysis for modeling spatio-temporal data

Master's Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology in the Degree Programme in Engineering Physics.

Espoo, November 29, 2009

Supervisor: Professor Erkki Oja
Instructor: Ph.D. Alexander Ilin

Author: Jaakko Luttinen
Degree programme: Degree Programme in Engineering Physics
Major subject: Computer and Information Science
Minor subject: Computational Science and Engineering

Title: Gaussian-process factor analysis for modeling spatio-temporal data

Title in Finnish: Gaussisiin prosesseihin perustuva faktorianalyysimalli avaruusajalliselle datalle

Chair: T-61 Computer and Information Science

Supervisor: Professor Erkki Oja

Instructor: Ph.D. Alexander Ilin

The main theme of this thesis is analyzing and modeling large spatio-temporal datasets, such as global temperature measurements. The task is typically to extract relevant structure and features for predicting or studying the system. This can be a challenging problem because simple models are often not able to capture the complex structure suffiently well, and more sophisticated models can be computationally too expensive in practice.

This thesis presents a novel spatio-temporal model which extends factor analysis by setting Gaussian process priors over the spatial and temporal components. In contrast to factor analysis, the presented model is capable of modeling complex spatial and temporal structure. Compared to standard Gaussian process regression over the spatio-temporal domain, the presented model gains substantial computational savings by operating only in the spatial or temporal domain at a time. Thus, it is feasible to model larger spatio-temporal datasets than with standard Gaussian process regression.

The new model combines the modeling assumptions of several traditional techniques used for analyzing spatially and temporally distributed data: kriging is used for modeling spatial dependencies; empirical orthogonal functions reduce the dimensionality of the problem; and temporal smoothing finds relevant features from time series.

The model is applied to reconstruct missing values in a historical sea surface temperature dataset. The results are promising and suggest that the proposed model may outperform the state-of-the-art reconstruction systems.

Number of pages: 73 **Keywords:** factor analysis, Gaussian processes, variational Bayesian inference

Faculty fills
Approved: Library code:

Tekijä: Jaakko Luttinen
Koulutusohjelma: Teknillisen fysiikan koulutusohjelma
Pääaine: Informaatiotekniikka
Sivuaine: Laskennallinen tiede ja tekniikka

Työn nimi: Gaussisiin prosesseihin perustuva faktorianalyysimalli avaruusajalliselle datalle

Title in English: Gaussian-process factor analysis for modeling spatio-temporal data

Professuuri: T-61 Informaatiotekniikka
Työn valvoja: Professori Erkki Oja
Työn ohjaaja: TkT Alexander Ilin

Tämän työn aiheena on suurien avaruusajallisten datakokoelmien, kuten maailmanlaajuisten lämpötilamittausten, analyysi ja mallinnus. Tehtävänä on yleensä löytää keskeisiä rakenteita ja piirteitä, joita voitaisiin hyödyntää systeemin käyttäytymisen ennustamiseen tai tutkimiseen. Tämä voi kuitenkin olla haastavaa, sillä yksinkertaiset mallit eivät kykene löytämään monimutkaisia rakenteita riittävän hyvin ja monimutkaisemmat mallit voivat olla laskennallisesti liian raskaita.

Työssä esitellään uusi avaruusajallinen malli, joka laajentaa faktorianalyysia asettamalla paikka- ja aikakomponenttien priorit gaussisilla prosesseilla. Toisin kuin faktorianalyysi, esitelty malli kykenee mallintamaan monimutkaisia paikka- ja aikarakenteita. Normaaliin gaussisten prosessien regressioon verrattuna malli saavuttaa merkittäviä säästöjä laskenta-ajoissa toimimalla kerrallaan vain joko paikan tai ajan suhteen. Täten menetelmä kykenee mallintamaan suurempia avaruusajallisia datakokoelmia kuin normaaleilla gaussisilla prosesseilla on yleensä mahdollista.

Uusi malli yhdistää oletuksia useista perinteisistä menetelmistä, joita käytetään paikan ja ajan suhteen jakautuneelle datalle: kriging-menetelmä mallintaa riippuvuuksia paikkojen välillä, empiiristen ortogonaalisten funktioiden menetelmä pienentää dimensionaalisuutta, ja ajallinen siloitus löytää aikasarjoista oleellisia piirteitä.

Mallia käytetään rekonstruoimaan puuttuvia arvoja historiallisessa meren lämpötilan datakokoelmassa. Tulokset ovat lupaavia ja antavat toiveita, että esitelty malli voi rekonstruoida paremmin kuin nykytason rekonstruointimenetelmät.

Sivumäärä: 73 **Avainsanat:** faktorianalyysi, gaussiset prosessit, variationaalinen bayesilainen päättely

Täytetään tiedekunnassa
Hyväksytty: Kirjasto:

Acknowledgements

This Master's thesis was done at the Department of Information and Computer Science at Helsinki University of Technology.

I would like to thank Professor Erkki Oja for the supervision and the chance to work in this inspiring environment. I also want to thank the people in the Bayes group for their encouragement and helpful discussions. I am deeply grateful to my instructor Dr. Alexander Ilin for suggesting this interesting research topic and generously sharing his time and views. I also wish to thank the people in room A322 for the distinctive working atmosphere.

I wish to thank my family and friends for giving me other things to think and experience. Especially, I want to thank my parents for their love.

Contents

Mathematical notation

lower- or upper-case letter	scalar, constant or scalar function
bold-face lower-case letter	column vector, vector-valued function
bold-face upper-case letter	matrix

The following notation and style for matrices in general:

$\mathbf{A}$	matrix
$\mathbf{a}_{:n}$	the n-th column vector of matrix $\mathbf{A}$
$\mathbf{a}_{m:}$	the m-th row vector of matrix $\mathbf{A}$ (as a column vector)
a_{mn}, $[\mathbf{A}]_{mn}$	the element on the m-th row and n-th column of matrix $\mathbf{A}$
$\mathbf{a}_{:}$, $[\mathbf{A}]_{:}$	all elements of matrix $\mathbf{A}$ as a column vector

The following notation and style for vectors in general:

$\mathbf{a}$	vector
a_m, $[\mathbf{a}]_m$	the m-th element of vector $\mathbf{a}$

$\langle\cdot\rangle$	expectation
$\mathbf{0}$	matrix or vector of zeros
$\mathbf{I}$	identity matrix
X	set of inputs
$\mathcal{X}$	input space
p	probability density function
q	approximate probability density function
$\mathrm{KL}\left(q\,\|p\right)$	the Kullback-Leibler divergence between two distributions q and p
$\mathcal{N}\left(\mathbf{y}\,|\boldsymbol{\mu},\boldsymbol{\Sigma}\right)$	Gaussian (normal) probability density function for variable $\mathbf{y}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{G}\left(\tau\,|a,b\right)$	gamma probability density function for variable τ with shape a and rate b
$\mathbf{f}$	vector of function values, $[\mathbf{f}]_i = f(\mathbf{x}_i)$
$\hat{\mathbf{f}}$	vector of auxiliary function values, $[\hat{\mathbf{f}}]_i = f(\hat{\mathbf{x}}_i)$
$k(\mathbf{x},\mathbf{x}';\boldsymbol{\theta})$	covariance function with parameters $\boldsymbol{\theta}$

$\mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}}$	matrix of covariances between $\mathbf{f}$ and $\hat{\mathbf{f}}$, $[\mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}}]_{ij} = k(\mathbf{x}_i, \hat{\mathbf{x}}_j; \boldsymbol{\theta})$
$\mathbf{K}_{\mathbf{f}}$	covariance matrix of $\mathbf{f}$, $[\mathbf{K}_{\mathbf{f}}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$
N	number of temporal inputs in a dataset
M	number of spatial inputs in a dataset
D	number of latent components
l_m	spatial locations in a dataset, $m = 1, \ldots, M$
t_n	time instances in a dataset, $n = 1, \ldots, N$
$\mathbf{Y}$	$M \times N$ matrix of data, $[\mathbf{Y}]_{mn} = y(l_m, t_n)$
$a_d(l)$	spatial feature function, $d = 1, \ldots, D$
$s_d(t)$	temporal feature function, $d = 1, \ldots, D$
$\mathbf{A}$	$M \times D$ matrix of loadings, $[\mathbf{A}]_{md} = a_d(l_m)$
$\mathbf{S}$	$D \times N$ matrix of factors or states, $[\mathbf{S}]_{dn} = s_d(t_n)$
$\hat{M}_d$	number of inducing inputs for the d-th spatial component
$\hat{N}_d$	number of inducing inputs for the d-th temporal component
$\hat{l}_{d\hat{m}}$	spatial inducing input, $\hat{m} = 1, \ldots, \hat{M}_d$
$\hat{t}_{d\hat{n}}$	temporal inducing input, $\hat{n} = 1, \ldots, \hat{N}_d$
$\hat{\mathbf{A}}$	set of auxiliary loadings, $[\hat{\mathbf{A}}]_{\hat{m}d} = a_d(\hat{l}_{\hat{m}})$
$\hat{\mathbf{S}}$	set of auxiliary factors or states, $[\hat{\mathbf{S}}]_{d\hat{n}} = s_d(\hat{t}_{\hat{n}})$
τ	precision (i.e., inverse variance) of noise
$\mathcal{O}$	set of such indices (m, n) that the corresponding y_{mn} is observed
$\mathcal{O}_{:n}$	set of such indices m that the corresponding y_{mn} is observed
$\mathcal{O}_{m:}$	set of such indices n that the corresponding y_{mn} is observed

List of Abbreviations

AR	autoregressive
BSS	blind source separation
CV	cross-validation
EM	expectation maximization
EOF	empirical orthogonal function (analysis)
FA	factor analysis
GP	Gaussian process
ICA	independent component analysis
KL	Kullback-Leibler (divergence)
LVM	latent variable model
MCMC	Markov chain Monte Carlo
PCA	principal component analysis
RBF	radial basis function (network)
SVD	singular value decomposition
VB	variational Bayesian

Chapter 1

Introduction

1.1 Problem setting

A spatio-temporal dataset is a collection of measurements which are taken at several locations at different times. Examples of such data include temperature and air pressure in climatology, apartment prices in economics as well as mortality rates in medical science. The data has typically some underlying dynamics, which can be extremely complex. In addition, these datasets can be large in size.

Analysis and modeling of such large and complex spatio-temporal datasets is a challenging task. For instance, the climate system has been studied by using both physical and statistical models. Physical modeling utilizes deep understanding of the first principles, leading to complex and computationally demanding simulations. Although physical modeling remains indispensable, statistical models offer a more practical approach to the problem. They can be applied to climate observations to study complex climate variability, which can significantly contribute to the climate knowledge.

Statistical analysis of complex models can be done in a principled way by using Bayesian methods. This framework makes it straightforward to perform several important tasks, including predicting unobserved variables, quantifying uncertainty in the predictions, comparing different models and handling missing data. However, Bayesian methods often lead to intractable integrals and high computational cost, thus some approximation techniques must be used to solve these issues.

For modeling and exploring high-dimensional systems, factor analysis (FA) gives a reasonable basis. The model assumes that the observed variables are generated as a linear combination of latent sources. These latent components give insight into the observed data variability and make it possible to predict missing data. However, standard FA is limited because not all prior informa-

tion is used. For example, FA discards information about temporal correlations in the data. Therefore, it is often necessary to extend the basic model in order to model the system appropriately.

Spatial and temporal prior knowledge can be incorporated into FA using Gaussian processes (GP). They enable setting flexible priors over functions, for instance, whether the function is stationary, smooth, or roughly periodic. By setting GP priors over the spatial and temporal patterns in FA, the components become continuous functions and can be predicted at arbitrary locations at any time. However, GPs do not scale well to large problems due to high computational cost. This computational limit can be pushed further by using sparse approximations.

1.2 Contributions of the thesis

This thesis presents a novel extension of factor analysis for exploratory analysis and modeling of spatio-temporal data. The model sets Gaussian process priors over the spatial and temporal components in factor analysis. Preliminary results with the model are published in a conference paper (Luttinen and Ilin, 2009).

From a modeling viewpoint, the model achieves significant advantages over standard Gaussian process regression and factor analysis approaches by combining them. In contrast to factor analysis, the presented model is capable of modeling complex spatial and temporal structure. Compared to the standard Gaussian process regression over the spatio-temporal domain, the presented model gains substantial computational savings by operating only in the spatial or temporal domain at a time. Thus, modeling of very large datasets which have been infeasible for standard Gaussian process regression become feasible for the proposed model.

The new model combines the modeling assumptions of several traditional techniques used for analysis of spatially and temporally distributed data: kriging is used for modeling spatial dependencies; empirical orthogonal functions reduce the dimensionality of the problem; and temporal smoothing finds relevant features from time series by removing noise.

The model is applied to reconstruct missing values in a historical sea surface temperature dataset. The results are promising and suggest that the proposed model may outperform the state-of-the-art reconstruction systems.

1.3 Structure of the thesis

The thesis is organized as follows: Chapter 2 presents the Bayesian framework for doing inference in a principled way. As this often leads to analytically

intractable integrals, variational Bayesian methods are introduced for finding approximate solutions efficiently. Chapter 3 presents probabilistic latent variable models for modeling high-dimensional datasets. It is shown how the basic factor analysis model can be extended to handle dynamics and nonlinearities in the data. Chapter 4 defines Gaussian processes for performing nonlinear regression and presents the state-of-the-art sparse approximation method for inference on large datasets. Chapter 5 introduces the novel Bayesian model as a combination of factor analysis and Gaussian processes. For learning the model, efficient algorithms are presented by using different levels of variational Bayesian approximations. Chapter 6 presents experimental results with the model using artificially generated data and a historical sea surface temperature dataset. Chapter 7 concludes the thesis and discusses some directions for future work. The appendices provide some technical details for interested readers.

Chapter 2

Bayesian inference

The Bayesian framework gives a principled way of doing modeling and data analysis. The state of knowledge is presented by probabilities, and thus the simple rules of probability theory can be used for doing inference. The same basic rules are used regardless of the complexity or the application field of the problem.

Bayesian modeling has several advantages over ad hoc approaches including: 1) The probabilities account for the uncertainty in the results. 2) Missing values are usually not a problem because the whole framework is about incomplete knowledge. 3) Model comparison can be done in a principled way. 4) Overfitting is prevented by combining many models. 5) Modeling assumptions and priors are expressed explicitly and can be altered. 6) Existing models can be straightforwardly modified, extended or used as a building block for more complex models.

This chapter gives a brief introduction to Bayesian modeling. Section 2.1 explains how Bayesian inference can be interpreted as a unique system of consistent rational reasoning under uncertainty. Section 2.2 shows how to apply this framework to modeling problems. However, modeling can rarely be performed exactly, thus some approximation methods are needed, as discussed in Section 2.3. The section also explains in more details the variational Bayesian approximation methods, and Section 2.4 illustrates the approximation in practice by using a simple toy example. More detailed philosophical and machine-learning oriented views can be found in, for instance, the books by Jaynes (2003) and Bishop (2006), respectively.

2.1 Probability theory

The probability theory can be seen as common sense reduced to calculation. Jaynes (2003) presented a fascinating derivation of the probability theory,

starting from very basic qualitative assumptions about rational reasoning:

(I) Degrees of plausibility are represented by real numbers.

(II) The theory qualitatively agrees with common sense.

(III) Consistent reasoning:

(IIIa) If a conclusion can be reasoned out in more than one way, then every possible way must lead to the same result.

(IIIb) Always take into account all the evidence relevant to a question.

(IIIc) Always represent equivalent states of knowledge by equivalent plausibility assignments.

After long and rigorous derivations, the resulting unique quantitative rules are the well-known product rule

$$p(A, B) = p(A|B)p(B) = p(B|A)p(A) \tag{2.1}$$

and the sum rule

$$p(A) + p(\overline{A}) = 1,$$

where A and B are hypotheses, and $\overline{A}$ is the complement of A. The probabilities $p(\cdot)$ represent the state of knowledge, where certainty is represented by 1 and impossibility by 0. Therefore, applying the Bayesian probability theory to inference problems is nothing but using common sense consistently.

2.2 Probabilistic modeling

In Bayesian modeling, an unknown system is learned given some observations $\mathbf{y}$. The prior beliefs about the system are formed into a likelihood function $p(\mathbf{y}|\mathrm{Z}, \mathcal{M})$, where Z is some latent (i.e., unknown) variable and $\mathcal{M}$ represents modeling assumptions. The prior belief for Z is expressed in the form $p(\mathrm{Z}|\mathcal{M})$. The posterior probability is obtained by applying the Bayes' rule

$$p(\mathrm{Z}|\mathbf{y}, \mathcal{M}) = \frac{p(\mathbf{y}|\mathrm{Z}, \mathcal{M})p(\mathrm{Z}|\mathcal{M})}{p(\mathbf{y}|\mathcal{M})}, \tag{2.2}$$

which follows from the product rule (2.1) (see, e.g., Gelman et al., 2003). The denominator $p(\mathbf{y}|\mathcal{M})$ is called the marginal likelihood, defined as

$$p(\mathbf{y}|\mathcal{M}) = \int p(\mathbf{y}|\mathrm{Z}, \mathcal{M})p(\mathrm{Z}|\mathcal{M})d\mathrm{Z},$$

which is the probability of the observations assuming the model $\mathcal{M}$. Thus, it can be used for model selection by evaluating the marginal likelihood of several models and choosing the model with the highest marginal likelihood. Typically, the conditioning on the model is not explicitly shown if there is no risk of misunderstanding. Thus, we discard $\mathcal{M}$ from our notation.

In order to simplify the calculations, the prior $p(\mathrm{Z})$ is often chosen to be of such a form that the resulting posterior $p(\mathrm{Z}|\mathbf{y})$ distribution is in the same family as the prior. This type of prior distribution is called a conjugate prior for the likelihood.

As an example, we present the conjugate priors for the multivariate normal (also known as Gaussian) distribution

$$p(\mathbf{y}|\mathbf{f}, \mathbf{\Sigma_y}) = \mathcal{N}\left(\mathbf{y}\,|\mathbf{f}, \mathbf{\Sigma_y}\right) = (2\pi)^{-\frac{N}{2}} |\mathbf{\Sigma_y}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{y}-\mathbf{f})^{\mathrm{T}}\mathbf{\Sigma_y}^{-1}(\mathbf{y}-\mathbf{f})\right],$$

where N is the length of the vector $\mathbf{y}$, $\mathbf{f}$ is a $N \times 1$ mean vector and $\mathbf{\Sigma_y}$ is a $N \times N$ covariance matrix. Here, the latent variables are $\mathrm{Z} = \{\mathbf{f}, \mathbf{\Sigma_y}\}$. For more details, refer to, for instance, the book by Gelman et al. (2003).

The conjugate prior of the mean variable $\mathbf{f}$ is also a Gaussian

$$p(\mathbf{f}) = \mathcal{N}\left(\mathbf{f}\,|\boldsymbol{\mu}, \mathbf{\Sigma_f}\right),$$

which results in the posterior Gaussian distribution

$$p(\mathbf{f}|\mathbf{y}) = \mathcal{N}\left(\mathbf{f}\,\middle|(\mathbf{\Sigma_f}^{-1} + \mathbf{\Sigma_y}^{-1})^{-1}(\mathbf{\Sigma_f}^{-1}\boldsymbol{\mu} + \mathbf{\Sigma_y}^{-1}\mathbf{y}), (\mathbf{\Sigma_f}^{-1} + \mathbf{\Sigma_y}^{-1})^{-1}\right). \tag{2.3}$$

The posterior mean of $\mathbf{f}$ is a weighted average of the prior mean $\boldsymbol{\mu}$ and the data $\mathbf{y}$. The posterior inverse covariance matrix is the sum of the inverted covariance matrices in the prior and likelihood.

The conjugate prior of the covariance matrix is a bit more complicated. Let us consider an isotropic covariance matrix of form $\mathbf{\Sigma_y} = \tau^{-1}\mathbf{I}$. The conjugate prior of τ is the gamma distribution

$$p(\tau) = \mathcal{G}\left(\tau\,|a, b\right) = \frac{b^a}{\Gamma(a)}\tau^{a-1}e^{-bx}, \quad \tau > 0,$$

where a and b are called a shape and rate parameters, respectively, and Γ is the gamma function. The resulting posterior gamma distribution equals

$$p(\tau|\mathbf{y}) = \mathcal{G}\left(\tau\,\middle|\alpha + \frac{1}{2}N, \beta + \frac{1}{2}\sum_{n=1}^{N}(y_n - f_n)^2\right).$$

The distributions have the following expectations. A Gaussian variable $\mathbf{f} \sim \mathcal{N}\left(\boldsymbol{\mu}, \mathbf{\Sigma_f}\right)$ has moments

$$\begin{aligned}\langle\mathbf{y}\rangle &= \boldsymbol{\mu}\\ \left\langle\mathbf{y}\mathbf{y}^{\mathrm{T}}\right\rangle &= \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}} + \mathbf{\Sigma A}.\end{aligned}$$

A variable $\tau \sim \mathcal{G}(a, b)$ has expectations

$$\langle \tau \rangle = \frac{a}{b}$$
$$\langle \log \tau \rangle = \psi(a) - \log b,$$

where ψ is the digamma function. These expectations are used in our model.

2.3 Posterior approximations

After data $\mathbf{y}$ is given, it is a central task in Bayesian data analysis to compute the posterior distribution $p(\mathrm{Z}|\mathbf{y})$ of the unknown variables $\mathrm{Z} = \{\mathrm{Z}_1, \ldots, \mathrm{Z}_M\}$. However, usually the posterior distribution (2.2) includes integrals that are analytically intractable or computationally too heavy. Therefore, one has to resort to some approximation methods, which can roughly be divided into two categories: stochastic and deterministic techniques (Bishop, 2006).

Stochastic techniques approximate the posterior distribution with a finite number of samples. The samples from the intractable posterior may be obtained in several ways depending on the problem. These stochastic techniques are covered comprehensively in the book by Gelman et al. (2003). In complex problems, sampling is often implemented with random-walk type of algorithms, called Markov chain Monte Carlo (MCMC). In general, stochastic methods have the property that the approximation approaches the true posterior at the limit of infinite computation time. However, for large and complex problems, the convergence can be extremely slow.

Deterministic methods use analytic approximations to the posterior. The resulting approximate distribution is often evaluated efficiently, but it usually requires extra work because some formulas must be derived analytically. The approximate distribution does not, in general, recover the true posterior distribution exactly. Important deterministic approximations include: maximum likelihood and maximum a posteriori methods, which approximate the posterior distribution with a point estimate; Laplace method, which fits a Gaussian distribution to a mode of the posterior probability density function; and variational Bayesian (Jordan et al., 1998) and expectation propagation (Minka, 2001) methods, which find a simple distribution by minimizing information-theoretic distance to the true distribution. The following subsections give more details on the relevant approximation methods for the thesis: maximum likelihood, maximum a posteriori and variational Bayesian methods.

2.3.1 Maximum likelihood and maximum a posteriori

Maximum likelihood (ML) and maximum a posteriori (MAP) estimates of a variable Z are defined as (see, e.g., Bishop, 2006)

$$\mathrm{Z}_{\mathrm{ML}} = \arg\max_{\mathrm{Z}} \{\log p(\mathbf{y}|\mathrm{Z})\},$$
$$\mathrm{Z}_{\mathrm{MAP}} = \arg\max_{\mathrm{Z}} \{\log p(\mathrm{Z}|\mathbf{y})\} = \arg\max_{\mathrm{Z}} \{\log p(\mathbf{y}|\mathrm{Z}) + \log p(\mathrm{Z})\}.$$

Both estimates summarize the distribution with a single point, ignoring all the uncertainty in the variable Z. Although this may lead to badly overfitted approximations, they can be sufficient and accurate for sharply peaked unimodal posterior distributions. ML and MAP estimates also have the advantage that they can be found efficiently using standard optimization methods.

2.3.2 Variational Bayesian methods

In variational Bayesian (VB) methods (see, e.g., Bishop, 2006) the idea is often to find an approximate distribution $q(\mathrm{Z}) \approx p(\mathrm{Z}|\mathbf{y})$ which minimizes the Kullback-Leibler (KL) divergence of $p(\mathrm{Z}|\mathbf{y})$ from $q(\mathrm{Z})$, defined as

$$\mathrm{KL}(q||p) = -\int q(\mathrm{Z}) \log \frac{p(\mathrm{Z}|\mathbf{y})}{q(\mathrm{Z})} d\mathrm{Z}.$$

The divergence is always nonnegative and zero only when $q(\mathrm{Z}) = p(\mathrm{Z}|\mathbf{y})$. However, it can not be evaluated if the true posterior distribution is intractable. Thus, the optimization has to be performed indirectly.

It turns out that the KL divergence can be minimized by maximizing a specific lower bound of the marginal likelihood. To begin with, the log marginal likelihood is decomposed as

$$\begin{aligned}
\log p(\mathbf{y}) &= \log \frac{p(\mathbf{y}, \mathrm{Z})}{p(\mathrm{Z}|\mathbf{y})} \\
&= \log \frac{p(\mathbf{y}, \mathrm{Z}) q(\mathrm{Z})}{p(\mathrm{Z}|\mathbf{y}) q(\mathrm{Z})} \\
&= \int q(\mathrm{Z}) \log \frac{p(\mathbf{y}, \mathrm{Z}) q(\mathrm{Z})}{p(\mathrm{Z}|\mathbf{y}) q(\mathrm{Z})} d\mathrm{Z} \\
&= \mathcal{L}(q) + \mathrm{KL}\left(q \,\|p\right),
\end{aligned}$$

where we have defined

$$\mathcal{L}(q) = \int q(\mathrm{Z}) \log \frac{p(\mathbf{y}, \mathrm{Z})}{q(\mathrm{Z})} d\mathrm{Z}. \tag{2.4}$$

Because $\log p(\mathbf{y})$ is constant with respect to q, changes in q result in opposite changes to $\mathcal{L}(q)$ and $\mathrm{KL}(q \,\|p)$. Thus, the Kullback-Leibler divergence can be minimized by maximizing $\mathcal{L}(q)$. Furthermore, because the KL divergence is always nonnegative, the $\mathcal{L}(q)$ can be seen as a lower bound of the log marginal likelihood, that is,

$$\log p(\mathbf{y}) \geq \mathcal{L}(q).$$

This lower bound can be used for model selection or comparison similarly as the true log marginal likelihood.

Actually, there is not yet anything approximate in this optimization, because clearly the optimal solution is the true posterior distribution itself $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{Y})$. In order to find a tractable solution, the range of functions is typically restricted somehow. On the other hand, the range must be as rich and flexible as possible in order to find a good approximation. This can be achieved by assuming a specific form for the distribution.

We restrict the class of approximate distributions by assuming that the q distribution factorizes with respect to some grouping of the variables, that is,

$$q(\mathrm{Z}) = \prod_{m=1}^{M} q_m(Z_m).$$

Inserting this distribution to the lower bound (2.4) and organizing the terms with respect to the m-th group Z_m results in

$$\begin{aligned}\mathcal{L}(q) &= \int \cdots \int \left(\prod_{m=1}^{M} q_m(Z_m) \right) \log \frac{p(\mathbf{y}, \mathrm{Z})}{\prod_{n=1}^{M} q_n(Z_n)} dZ_1 \cdots dZ_M \\ &= \int q_m(Z_m) \left(\log \tilde{p}(\mathbf{y}, Z_m) - \log q(Z_m) \right) dZ_m + \text{const}, \end{aligned} \tag{2.5}$$

where const represents terms that are constant with respect to $q_m(Z_m)$ and

$$\begin{aligned}\log \tilde{p}(\mathbf{y}, Z_m) &= \int \cdots \int \left(\prod_{n \neq m} q_n(Z_n) \right) \log p(\mathbf{y}, \mathrm{Z}) d\mathrm{Z}_{\setminus m} \\ &= \langle \log p(\mathbf{y}, \mathrm{Z}) \rangle_{\setminus m}, \end{aligned}$$

where the expectation $\langle \cdot \rangle$ is taken over the q distributions of all the other variables except Z_m. Equation (2.5) can be seen as the Kullback-Leibler divergence between $q_m(Z_m)$ and $\tilde{p}(\mathbf{y}|Z_m)$

$$\mathcal{L}(q) = \int q_m(Z_m) \log \frac{\tilde{p}(\mathbf{y}, Z_m)}{q(Z_m)} dZ_m + \text{const},$$

and it follows that the optimal $q_m(Z_m)$ satisfies

$$q_m(Z_m) \propto \tilde{p}(\mathbf{y}, Z_m) = \exp\left(\langle \log p(\mathbf{y}, \mathrm{Z}) \rangle_{\setminus m}\right). \tag{2.6}$$

Thus, the log marginal likelihood lower bound (2.4) can be maximized with respect to one factor $q_m(Z_m)$ at a time. In order to maximize the approximate joint distribution $q(\mathrm{Z})$, the individual factors can be updated in turns by iterating until convergence. This learning method is called the variational Bayesian expectation maximization (VB-EM) algorithm (Attias, 2000; Beal and Ghahramani, 2003). Alternatively, it is also possible to use, for instance, gradient-based optimization methods to optimize the parameters of the approximate q distributions (see, e.g., Honkela et al., 2008).

In the end, one has approximations of the posterior marginal distributions of each variable group:

$$p(\mathrm{Z}_m|\mathbf{y}) \approx q(\mathrm{Z}_m),$$

and the joint posterior distribution:

$$p(\mathrm{Z}|\mathbf{y}) \approx q(\mathrm{Z}).$$

The lower bound $\mathcal{L}(q)$ for the log marginal likelihood is given in (2.4).

2.4 Toy example

We present a simple toy example to illustrate the variational Bayesian approximation. The toy data consists of a single observation $y = 1$ modeled as a product of two unknown variables a and s with additive noise:

$$y = as + \text{noise}. \tag{2.7}$$

The priors for the variables a and s are given as Gaussian distributions

$$p(a) = \mathcal{N}(a\,|0, 1), \qquad p(s) = \mathcal{N}(s\,|0, 1).$$

The noise in (2.7) is assumed to be Gaussian, resulting in a likelihood function

$$p(y|a, s) = \mathcal{N}\left(y\,\middle|as, \sigma^2\right),$$

where the variance σ^2 is fixed to 0.1 for simplicity.

The true posterior is approximated with a factorized distribution as

$$p(a, s|y) \approx q(a, s) = q(a)q(s).$$

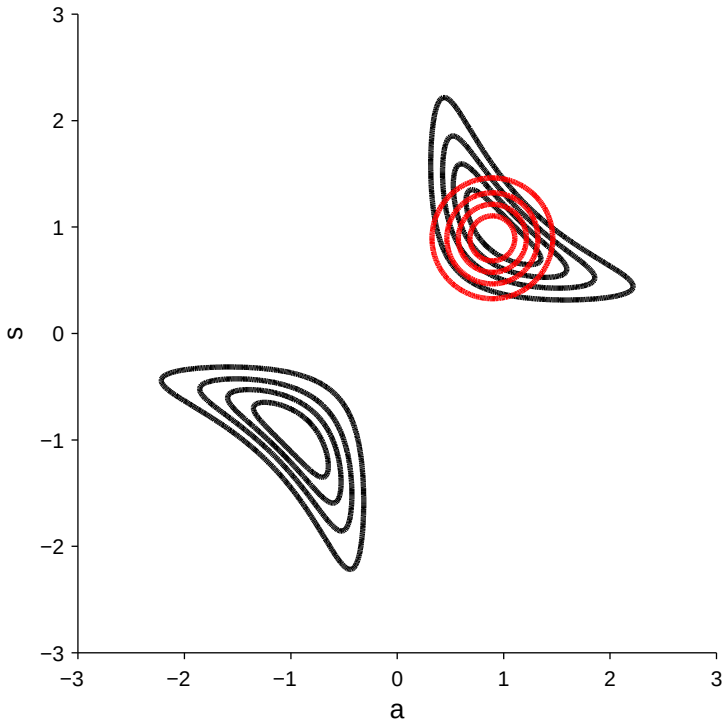


Figure 2.1: The true posterior $p(a, s|y)$ is shown as black contours, and the approximate posterior $q(a, s)$ as red contours.

This approximation can be optimized by using the VB-EM algorithm, which consists of alternate updates of the factors $q(a)$ and $q(s)$. The update rules for these factors can be evaluated by applying the general update rule (2.6), resulting in

$$q(a) = \mathcal{N}\left(a \,|\bar{a}, \sigma_a^2\right),$$

where the parameters are defined as

$$\begin{aligned} \sigma_a^2 &= (1 + \sigma^{-2} \left\langle s^2 \right\rangle)^{-1}, \\ \bar{a} &= \sigma_a^2 \sigma^{-2} \left\langle s \right\rangle y, \end{aligned}$$

and the expectations $\langle s^2 \rangle$ and $\langle s \rangle$ are evaluated with respect to $q(s)$. The factor $q(s)$ is updated by using identical formulas where a and s have been appropriately exchanged. The update rules are applied until convergence.

Figure 2.1 shows the contours of the true posterior $p(a, s|y)$ and the approximation $q(a, s)$. The approximate distribution $q(a, s)$ has converged to one of the two modes and captured the probability mass in that mode rather well. However, the approximation discards some posterior correlations because

the variables a and s were assumed to be independent a posteriori. This behavior is a general feature of the VB methods that factorize the approximate distribution with respect to the variables.

2.5 Conclusions

This chapter presented the Bayesian framework, which gives a principled and consistent way of doing inference. The basics of applying the framework to probabilistic modeling were given. As the Bayesian inference often leads to analytically intractable integrals, we discussed some approximation methods. The variational Bayesian methodology was explained in more detail and illustrated with a simple toy example. This example showed the general properties of the factorized VB posterior approximations: they often capture only one mode of the true distribution and discard posterior correlations between the variables.

Chapter 3

Latent variable models

Latent variable models (LVM) is a general tool for analyzing and modeling large high-dimensional datasets. They can be used for reducing dimensionality and finding important characteristics from datasets by explaining the data with lower dimensional latent features. Many LVMs can be seen as extensions of the basic factor analysis model, which has a significant role in exploratory analysis, although it is a simple linear model.

This chapter presents latent variable models which work as a baseline for further development of spatio-temporal factor analysis in Chapter 5. Section 3.1 defines principal component analysis mathematically. Section 3.2 formulates it as a probabilistic model and extends it to factor analysis. Section 3.3 explains some dynamic extensions, and Section 3.4 discusses the problem of finding meaningful latent sources from data.

3.1 Principal component analysis

Principal component analysis (PCA) is a classical method for dimensionality reduction (Jolliffe, 2002). In environmental statistics, PCA is known as the method of empirical orthogonal function (EOF) analysis (von Storch and Zwiers, 1999; Finkenstädt et al., 2007). These methods find uncorrelated components that explain as much variance in the data as possible. Thus, they can be seen to extract dominant patterns which can give insight to the high-dimensional data.

PCA can be derived, for instance, from the minimization of the mean squared error. Let us assume that the dataset consists of N data vectors $\mathbf{y}_{:1}, \ldots, \mathbf{y}_{:N}$ with dimensionality M, forming $M \times N$ matrix $\mathbf{Y} = [\mathbf{y}_{:1}, \ldots, \mathbf{y}_{:N}]$. As a preprocessing step, the row-wise mean is removed from $\mathbf{Y}$. Then, $\mathbf{Y}$ is decomposed as $\mathbf{Y} \approx \mathbf{AS}$, where $\mathbf{A}$ is an $M \times D$ matrix of loadings and $\mathbf{S}$ is a $D \times N$ matrix of factors. The dimensionality D is chosen such that $D < M$

and $D < N$. The task is to find such $\mathbf{A}$ and $\mathbf{S}$ that minimize the reconstruction error

$$E_{\mathbf{AS}} = \|\mathbf{Y} - \mathbf{AS}\|^2 = \sum_{m=1}^{M}\sum_{n=1}^{N}(y_{mn} - \sum_{d=1}^{D} a_{md}s_{dn})^2. \tag{3.1}$$

The subspace spanned by the columns of $\mathbf{A}$ is called the principal subspace. Note that $\mathbf{A}$ can be rotated arbitrarily by compensating it in $\mathbf{S}$ as $\mathbf{AS} = (\mathbf{AR})(\mathbf{R}^{-1}\mathbf{S})$. Therefore, without loss of generality, in order to find a unique solution, one can require that the column vectors of $\mathbf{A}$ are mutually orthogonal, and the row vectors of $\mathbf{S}$ are also mutually orthogonal and scaled to unit variance. If these additional requirements are used, the method is called principal component analysis.

If the data has no missing values, the principal components can be found by using the singular value decomposition (SVD)

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}, \tag{3.2}$$

where $\mathbf{U}$ is an $M \times M$ orthogonal matrix, $\mathbf{V}$ is an $N \times N$ orthogonal matrix, and $\mathbf{\Sigma}$ is an $M \times N$ matrix with the singular values on the main diagonal. The PCA solution corresponds to selecting the D largest singular values, and forming $\mathbf{A}$ and $\mathbf{S}$ from the corresponding D columns of $\frac{1}{\sqrt{N}}\mathbf{U}\mathbf{\Sigma}$ and D rows of $\sqrt{N}\mathbf{V}^{\mathrm{T}}$, respectively.

However, standard PCA has some limitations as there is, for example, no unified way of handling missing values nor selecting the correct number of principal components. Although missing values can be handled by adapting algorithms based on the minimization of the cost function (3.1), this may lead to major overfitting problems for sparse datasets (Ilin and Raiko, 2008). The number of principal components can be chosen by using, for instance, cross-validation (CV) in which the data is split into several sets, and the choice is made based on the generalization performance over each set by using the other sets for estimating the variables $\mathbf{A}$ and $\mathbf{S}$ (Jolliffe, 2002). However, estimating the variables several times can be computationally intensive for large datasets. These issues can easily be solved by adopting the probabilistic framework. This framework also makes it straightforward to extend the model to more complex modeling problems.

3.2 Factor analysis

Factor analysis (FA) can be interpreted as the following probabilistic latent variable model (see, e.g., Bishop, 1999a):

$$\mathbf{y}_{:n} = \mathbf{A}\mathbf{s}_{:n} + \boldsymbol{\mu} + \boldsymbol{\varepsilon}_{:n}, \qquad n = 1, \ldots, N, \tag{3.3}$$

where $\mathbf{A}$ is the $M \times D$ loading matrix, $\mathbf{s}_{:n}$ are the columns of the $D \times N$ matrix of factors, $\boldsymbol{\mu}$ is a $M \times 1$ bias term and $\boldsymbol{\varepsilon}_{:n}$ is a $M \times 1$ vector of noise. The priors for the latent variables $\mathbf{s}_{:n}$ and $\boldsymbol{\varepsilon}_{:n}$ are given as

$$p(\mathbf{s}_{:n}) = \prod_{d=1}^{D} \mathcal{N}\left(s_{dn} \left| 0, 1\right.\right), \qquad p(\boldsymbol{\varepsilon}_{:n}) = \prod_{m=1}^{M} \mathcal{N}\left(\varepsilon_{mn} \left| 0, \tau_m^{-1}\right.\right), \tag{3.4}$$

where τ_m^{-1} is the noise variance in the m-th dimension.

Maximum likelihood estimates for the parameters $\mathbf{A}$, $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$ can be found by using the expectation maximization (EM) algorithm (Dempster et al., 1977). The algorithm iteratively alternates between computing the expectation of the log likelihood with respect to the current estimate of the posterior distribution (E-step) and maximizing this expectation with respect to the parameters (M-step). PCA is a special case of FA which assumes isotropic noise, that is, $\tau_m = \tau$, and it can be shown that the maximum likelihood estimation yields the result of standard PCA in the limit $\tau \to \infty$ (Tipping and Bishop, 1999).

Instead of using ML estimates, it is possible to set priors for $\mathbf{A}$, $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$ and approximate the joint posterior distribution without using point estimates. For instance, the priors could be

$$p(\mathbf{A}) = \prod_{m=1}^{M}\prod_{d=1}^{D} \mathcal{N}\left(a_{md} \left| 0, \alpha_d^{-1}\right.\right), \qquad p(\boldsymbol{\mu}) = \prod_{m=1}^{M} \mathcal{N}\left(\mu_m \left| 0, \beta\right.\right),$$
$$p(\boldsymbol{\alpha}) = \prod_{d=1}^{D} \mathcal{G}\left(\alpha_d \left| a_\alpha, b_\beta\right.\right), \qquad p(\boldsymbol{\tau}) = \prod_{m=1}^{M} \mathcal{G}\left(\tau_m \left| a_\tau, b_\tau\right.\right),$$

where α_d is used to automatically prune out irrelevant components, and the hyperparameters β, a_α, b_α, a_τ and b_τ can be fixed to small values (e.g., 10^{-3}) to obtain broad priors.

Inference with the model must be performed by using approximation methods because the joint posterior distribution $p(\mathbf{A}, \mathbf{S}, \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\tau}|\mathbf{Y})$ is intractable. Variational Bayesian methods can be used to approximate the distribution by a factorized distribution (Bishop, 1999b)

$$p(\mathbf{A}, \mathbf{S}, \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\tau}|\mathbf{Y}) \approx q(\mathbf{A})q(\mathbf{S})q(\boldsymbol{\mu})q(\boldsymbol{\alpha})q(\boldsymbol{\tau}).$$

The variables are strongly coupled in the true posterior, thus the VB-EM algorithm may suffer from zigzagging and converge slowly. To improve the rate of convergence, it is possible to apply some transformations which help in optimizing the factors jointly (see, e.g., Luttinen et al., 2009b).

3.3 Dynamic latent variable models

In some applications, the observations are time series $y_m(t)$ having some underlying temporal structure. Instead of modeling the dynamics of the time series $\mathbf{y}(t) = \begin{bmatrix} y_1(t) & \dots & y_M(t) \end{bmatrix}^{\mathrm{T}}$ directly, it can be more feasible to model the dynamics in some latent space by using latent time series $\mathbf{s}(t)$. The latent sources are then mapped to the observation space. Linear state-space models typically use linear dynamics in the latent space and a linear mapping from the latent space to the observations. Nonlinear state-space models can have nonlinear dynamics, a nonlinear mapping to the observations, or both.

3.3.1 Linear state-space models

A linear state-space model is defined as

$$\begin{aligned} \mathbf{y}(t) &= \mathbf{A}\mathbf{s}(t) + \text{noise}, \\ \mathbf{s}(t) &= \mathbf{B}\mathbf{s}(t-1) + \text{noise}, \end{aligned} \tag{3.5}$$

where $\mathbf{A}$ is a $M \times D$ loading matrix as in factor analysis, $\mathbf{B}$ is a $D \times D$ matrix describing the first-order autoregressive (AR) dynamics in the latent space, and the noise is often assumed to be Gaussian.

Learning this linear model has been studied extensively. If the parameters $\mathbf{A}$ and $\mathbf{B}$ are known, the latent states can be estimated by using the Kalman filter (Grewal and Andrews, 1993). If the parameters are unknown, maximum likelihood estimates can be found by utilizing the EM algorithm (Ghahramani and Roweis, 1999).

Linear state-space models have been applied in climate research extensively (see, e.g., Banerjee et al., 2004; Calder, 2007). Lopes et al. (2008), for instance, added spatial structure by using Gaussian processes[1]. Although AR models offer an efficient framework to model dynamics, low-order AR models may provide unrealistically simple dynamics, and the learning of high-order AR dynamics can be difficult. In addition, the samples are often required to be uniformly spaced.

3.3.2 Nonlinear state-space models

Nonlinear state-space models can be used for more complex dynamical structure. The model uses a nonlinear mapping $\mathbf{f}$ from the latent states $\mathbf{s}(t)$ to the observations $\mathbf{y}(t)$ and nonlinear dynamics $\mathbf{g}$ in the latent space:

$$\begin{aligned} \mathbf{y}(t) &= \mathbf{f}(\mathbf{s}(t)) + \text{noise}, \\ \mathbf{s}(t) &= \mathbf{g}(\mathbf{s}(t-1)) + \text{noise}, \end{aligned}$$

[1] Gaussian processes are discussed in the following chapter.

where the noise is often assumed to be Gaussian.

Learning a nonlinear state-space model is extremely difficult. Even if the nonlinearities $\mathbf{f}$ and $\mathbf{g}$ were known, they cause the posterior distribution of the states $\mathbf{s}(t)$ to be non-Gaussian. With unknown nonlinearities, the model becomes very flexible, thus some regularization is needed to prevent overfitting. In addition, there exist infinitely many solutions because any invertible nonlinear transformation of the states can be compensated by a suitable transformation of the mappings $\mathbf{f}$ and $\mathbf{g}$.

The nonlinearities $\mathbf{f}$ and $\mathbf{g}$ can be modeled in different ways. For instance, Roweis and Ghahramani (2001) modeled the nonlinearities with a radial basis function (RBF) network which was learned by using the EM algorithm. Valpola and Karhunen (2002) applied multi-layer perceptron (MLP) networks and used variational Bayesian methods for learning. The latent sources $s_d(t)$ were assumed to be independent in the VB posterior approximation, thus the algorithm favors solutions with decoupled latent space dynamics. Pang et al. (2007) modeled the nonlinear mapping $\mathbf{f}$ from the latent states $\mathbf{s}(t)$ to the observations $\mathbf{y}(t)$ with Gaussian processes but used a linear second-order AR model for the dynamics in the latent space. Park and Choi (2007), on the other hand, used a linear mapping $\mathbf{f}$ but a nonlinear mapping for the latent space dynamics $\mathbf{g}$ using Gaussian processes.

3.4 Blind source separation

In source separation problems, the goal is to find a set of meaningful sources $s_d(t)$ from a set of observation signals $y_d(t)$ which are mixtures of the sources. A classical example is the cocktail party problem, where several people are talking simultaneously in the same room and the task is to separate the voices of the different speakers by using recordings of several microphones in the same room. With minimum a priori assumptions about the sources, the source separation problem is called blind source separation (BSS).

Independent component analysis (ICA) is a tool to solve the BSS problem (Hyvärinen et al., 2001). It uses the linear latent variable model (3.3) but assumes that the latent components are non-Gaussian. The intuition behind the method can be understood from the central limit theorem, which states that the mean of a large number of independent random variables is approximately normally distributed under suitable conditions. If the observation signals are mixtures of independent sources, non-Gaussianity can be interpreted as a measure of independence for the latent sources. Thus, ICA algorithms typically find latent sources by maximizing their non-Gaussianity. However, standard ICA algorithms do not take into account any temporal structure in the observations.

If the observations are time series, the sources can be separated by exploiting the temporal structure. The separation is based on finding sources that have independent dynamics. This can be achieved by, for instance, decoupling the temporal correlations based on autocorrelations or frequency contents.

Separation based on autocorrelations can be performed by eliminating the latent source autocorrelations

$$\langle s_i(t)s_j(t-\delta)\rangle\,, \quad i \neq j$$

where δ is a time lag. The separation diagonalizes the sample covariance matrix $\frac{1}{T}\sum_{t=1}^{T}\mathbf{y}(t)\mathbf{y}(t)^{\mathrm{T}}$ and the estimate of the time-lagged covariance matrix $\frac{1}{T}\sum_{t=1}^{T}\mathbf{y}(t)\mathbf{y}(t-\tau)^{\mathrm{T}}$ (see, e.g., Tong et al., 1991). It is also possible to jointly diagonalize several time-lagged covariance matrices using different time lags τ. This approach is applied, for instance, in the algorithm called TDSEP (Ziehe and Müller, 1998).

Denoising source separation (DSS) constructs source separation algorithms by using denoising procedures (Särelä and Valpola, 2005). It looks for latent sources that are uncorrelated and maximize some desired properties, for example, smoothness, non-Gaussianity or distinct frequency structure. This framework makes it possible to incorporate prior knowledge about the sources because the desired property can be problem-specific. Exploiting the prior knowledge may help in finding a good representation of the data. For instance, Ilin et al. (2006) applied this framework in exploratory analysis of climate data to extract components with distinct frequency structure.

3.5 Conclusions

This chapter presented basic latent variable models. Factor analysis and principal component analysis are simple linear models for finding dominant patterns in data. They can be extended to dynamical models in order to take into account the temporal structure: linear state-space models offer possibly unrealistically simple dynamics, and nonlinear state-space models, on the other hand, are difficult to learn. With some extra criteria, these models can be used for finding independent and meaningful latent sources to solve the BSS problem.

Chapter 4

Gaussian process regression

In regression problems, the goal is to find an unknown function f which maps the input space $\mathcal{X}$ to the real-valued output space $\mathbb{R}$. Applications for regression analysis exist in almost every field: in climatology, the interest might be in modeling temperature as a function of time and location; in financial applications, one may attempt to predict the prices of apartments as a function of location, size and age.

Traditionally, the regression problem has been solved by setting some parameterized form for the function f and then learning the parameters, for instance, the bias and slope in linear regression or the coefficients in polynomial regression. After learning the parameters, predictions can be made conditioned only on the parameters. When such an approach is used, we can only model such functions that can be obtained with some values of the parameters. However, the real underlying phenomenon is probably much more complex than the parameterized function with a few degrees of freedom. Increasing the number of parameters increases the flexibility of the model, but the effect of the parameterization and the priors over the parameters become harder to interpret. Instead of working on a large set of parameters, one can work over the functions more directly by having interestingly an infinite number of parameters, which is called a non-parameteric approach.

Gaussian process (GP) is a non-parametric regression tool which allows the complexity of the function to increase as much as there is evidence in the data while keeping the model and the hyperparameters easily interpretable. Section 4.1 explains Gaussian processes, and Section 4.2 applies them to the regression problem. The prior assumptions about the unknown function are set in the form of covariance functions, which are discussed in Section 4.3 with some examples. The covariance functions include hyperparameters, which control the high-level properties of the unknown function and must be learned using some approximate methods, as presented in Section 4.4. Section 4.5 discusses the computational problem of applying Gaussian processes to large datasets and

presents the state-of-the-art variational Bayesian sparse approximations. For more detailed introduction to Gaussian processes, refer to, for instance, the book by Rasmussen and Williams (2006) or the shorter tutorial by MacKay (1998). When Gaussian processes are used for spatial interpolation, the approach is also known as kriging, which is a fundamental tool in the field of geostatistics (Cressie, 1993).

4.1 Introduction to Gaussian processes

Gaussian processes are used to set probability distributions over the space of functions $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is the input space. Therefore, the realizations of the random variables from this distribution are functions, and the standard tools of probabilistic inference can now be applied to functions. It is possible, for instance, to draw random functions, evaluate probabilities of given functions and use probabilistic calculus to obtain the posterior distribution over functions.

Gaussian process can be defined as a stochastic process $\{f(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$ for which any finite set of samples is normally distributed. For an arbitrary finite set of inputs $\mathrm{X} = \{\mathbf{x}_n \in \mathcal{X}\}_{n=1}^{N}$, the corresponding function values $\mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) & \dots & f(\mathbf{x}_N) \end{bmatrix}^{\mathrm{T}}$ are distributed as

$$\mathbf{f}|\mathrm{X} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}), \tag{4.1}$$

where the parameters $\boldsymbol{\mu}$ and $\mathbf{K}$ are defined as

$$\begin{aligned} [\boldsymbol{\mu}]_i &= m(\mathbf{x}_i), \\ [\mathbf{K}]_{ij} &= k(\mathbf{x}_i, \mathbf{x}_j), \end{aligned}$$

and the functions m and k are the mean and covariance functions, respectively. The mean function $m(\mathbf{x})$ describes what is the expected value of function f for input $\mathbf{x}$. The covariance function $k(\mathbf{x}, \mathbf{x}')$ defines the covariance between two function values $f(\mathbf{x})$ and $f(\mathbf{x}')$ based on the inputs $\mathbf{x}$ and $\mathbf{x}'$. Loosely speaking, the value of the covariance defines the expected similarity of the function values. For smooth functions, the function values for two nearby inputs are expected to be very similar, that is, their correlation should be close to one. The covariance functions are extremely flexible for setting rather high-level properties of the function, such as smoothness, periodicity and stationarity. They are the heart of Gaussian process modeling and will be discussed in more detail in Section 4.3. Learning with Gaussian processes is essentially learning the covariance function, but let us assume for now that the form of the covariance function $k(\mathbf{x}, \mathbf{x}')$ is given from some prior knowledge.

Defining the distribution for finite sets of function values is consistent, because Gaussian distribution has the property that marginalization of other variables does not affect the mean and covariance of the remaining variables. Thus, we can interpret (4.1) as a finite-dimensional marginal distribution of an infinite-dimensional Gaussian distribution, called Gaussian process. Refer to the paper by Orbanz (2009) for more detailed theoretical discussion on the existence, consistency and uniqueness of this infinite-dimensional probability distribution.

The infinite-dimensional Gaussian process distribution for a "random" function f is denoted as

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right). \tag{4.2}$$

As it represents the state of knowledge about the function, it can, similarly to other probability distributions in general, describe prior and posterior distributions. For a GP prior distribution, the mean function is usually assumed to be zero (i.e., $m(\mathbf{x}) = 0$) for simplicity.

4.2 Regression problem

One of the most common applications of Gaussian processes is the regression problem. The univariate regression model for input $\mathbf{x} \in \mathcal{X}$ and output $y \in \mathbb{R}$ is defined as

$$y = f(\mathbf{x}) + \text{noise}, \tag{4.3}$$

where $f : \mathcal{X} \to \mathbb{R}$ is the function of interest. The task is to estimate f from a known set of input-output pairs $\{y_n, \mathbf{x}_n\}_{n=1}^N$ called the training set.

The traditional approach parameterizes the function f with some regression parameters $\boldsymbol{\lambda}$, for instance,

$$f(x; \boldsymbol{\lambda}) = \lambda_1 + \lambda_2 x + \lambda_3 x^2.$$

The task is to learn the parameters $\boldsymbol{\lambda}$ given the data. In the Bayesian framework, this means finding the posterior distribution $p(\boldsymbol{\lambda}|\mathbf{y})$ of the regression parameters given some prior $p(\boldsymbol{\lambda})$. Then, the data can be discarded and predictions can be made based on the posterior distribution of $\boldsymbol{\lambda}$. However, the range of possible functions is limited to such functions that are obtained with some values of the regression parameters $\boldsymbol{\lambda}$. In addition, the prior beliefs about the function may be difficult to set as a prior distribution $p(\boldsymbol{\lambda})$.

In order to model complex non-linear functions flexibly, a Gaussian process can be used as the prior distribution over functions as in (4.2). If the mean

function is assumed to be zero a priori, the prior distribution of the function values $\mathbf{f}$ for the inputs $\mathrm{X} = \{\mathbf{x}_n\}_{n=1}^N$ equals

$$\mathbf{f}|\mathrm{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_\mathbf{f}), \tag{4.4}$$

where $\mathbf{K}_\mathbf{f}$ is the covariance matrix of the function values.

The notation for covariance matrices is as follows. Let $\{\mathbf{x}_n\}_{n=1}^N$ and $\{\mathbf{z}_m\}_{m=1}^M$ be two sets of inputs from the input space $\mathcal{X}$, and $\mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) & \dots & f(\mathbf{x}_N) \end{bmatrix}^\mathrm{T}$ and $\mathbf{g} = \begin{bmatrix} f(\mathbf{z}_1) & \dots & f(\mathbf{z}_M) \end{bmatrix}^\mathrm{T}$ the corresponding function values. The $N \times M$ matrix of the covariances between the function values $\mathbf{f}$ and $\mathbf{g}$ is denoted as $\mathbf{K}_{\mathbf{f},\mathbf{g}}$ and defined as $[\mathbf{K}_{\mathbf{f},\mathbf{g}}]_{ij} = k(\mathbf{x}_i, \mathbf{z}_j)$. We also use a shorthand notation $\mathbf{K}_\mathbf{f} \equiv \mathbf{K}_{\mathbf{f},\mathbf{f}}$. Note that although the notation of a covariance matrix $\mathbf{K}_{\mathbf{f},\mathbf{g}}$ uses the function values $\mathbf{f}$ and $\mathbf{g}$, the covariance matrix itself does not depend on the function values but on the corresponding inputs.

The noise term in the regression model (4.3) can be modeled with a Gaussian distribution, resulting in a likelihood function

$$\mathbf{y}|\mathbf{f}, \mathrm{X} \sim \mathcal{N}(\mathbf{f}, \boldsymbol{\Sigma}), \tag{4.5}$$

where $\mathbf{y} = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^\mathrm{T}$. Often, the noise covariance is assumed to be isotropic $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$. However, in Chapter 5 we will have to deal with non-isotropic, block-diagonal noise covariance matrices, thus the derivations here are done with an arbitrary noise covariance matrix $\boldsymbol{\Sigma}$.

The posterior distribution of the function values is obtained by applying the general posterior Gaussian distribution (2.3), yielding

$$\mathbf{f}|\mathrm{X}, \mathbf{y} \sim \mathcal{N}\left(\bar{\mathbf{f}}, \mathbf{V}_\mathbf{f}\right), \tag{4.6}$$

where

$$\bar{\mathbf{f}} = \left(\mathbf{K}_\mathbf{f}^{-1} + \boldsymbol{\Sigma}^{-1}\right)^{-1} \boldsymbol{\Sigma}^{-1}\mathbf{y}, \tag{4.7}$$

$$\mathbf{V}_\mathbf{f} = \left(\mathbf{K}_\mathbf{f}^{-1} + \boldsymbol{\Sigma}^{-1}\right)^{-1}. \tag{4.8}$$

The posterior predictive distribution of function values $\tilde{\mathbf{f}}$ for new inputs $\tilde{\mathrm{X}} = \{\tilde{\mathbf{x}}\}_{\tilde{n}=1}^{\tilde{N}}$ can be obtained by using the general equations for Gaussian distributions. The joint distribution of these new function values $\tilde{\mathbf{f}}$ and the observations $\mathbf{y}$ is

$$\begin{bmatrix} \mathbf{y} \\ \tilde{\mathbf{f}} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_\mathbf{f} + \boldsymbol{\Sigma} & \mathbf{K}_{\mathbf{f},\tilde{\mathbf{f}}} \\ \mathbf{K}_{\tilde{\mathbf{f}},\mathbf{f}} & \mathbf{K}_{\tilde{\mathbf{f}}} \end{bmatrix}\right).$$

Applying the formula of conditional Gaussian distribution (A.6) to this joint distribution results in the posterior predictive distribution

$$\tilde{\mathbf{f}}|\mathrm{X}, \mathbf{y}, \tilde{\mathrm{X}} \sim \mathcal{N}\left(\bar{\tilde{\mathbf{f}}}, \mathbf{V}_{\tilde{\mathbf{f}}}\right), \tag{4.9}$$

where

$$\bar{\tilde{\mathbf{f}}} = \mathbf{K}_{\tilde{\mathbf{f}},\mathbf{f}} \left(\mathbf{K}_{\mathbf{f}} + \boldsymbol{\Sigma}\right)^{-1} \mathbf{y}, \tag{4.10}$$

$$\mathbf{V}_{\tilde{\mathbf{f}}} = \mathbf{K}_{\tilde{\mathbf{f}}} - \mathbf{K}_{\tilde{\mathbf{f}},\mathbf{f}} \left(\mathbf{K}_{\mathbf{f}} + \boldsymbol{\Sigma}\right)^{-1} \mathbf{K}_{\mathbf{f},\tilde{\mathbf{f}}}. \tag{4.11}$$

Note that (4.9) can also be used to obtain the posterior distribution of the same function values $\tilde{\mathbf{f}} \equiv \mathbf{f}$, that is, $\tilde{\mathrm{X}} = \mathrm{X}$ and $\mathbf{K}_{\mathbf{f}} = \mathbf{K}_{\mathbf{f},\tilde{\mathbf{f}}} = \mathbf{K}_{\tilde{\mathbf{f}},\mathbf{f}} = \mathbf{K}_{\tilde{\mathbf{f}}}$. Then, applying the matrix inversion lemmas (A.2) and (A.1) to (4.10) and (4.11) gives the result in the same form as in (4.7) and (4.8). The equations for the parameters in (4.10) and (4.11) are computationally more feasible as they involve only one matrix inversion. However, we will not present formulas in the form which is computationally most efficient because the implementation is discussed separately in Appendix B.

Gaussian processes have a few advantages over the parameterized regression methods. First, Gaussian processes make the setting of priors over functions easily interpretable, whereas with parameterized regression models, the meaning and effect of the parameterization of the function f and the parameters $\boldsymbol{\lambda}$ can be difficult to understand. Second, parameterized regression models often correspond to GP models with some specific covariance functions and can thus be seen as special cases of the more general GP framework. For instance, AR models (3.5) can be seen as a special case of Gaussian processes which use a particular covariance function and the time instances t as inputs (Rasmussen and Williams, 2006).

4.3 Covariance functions

Covariance functions play an important role in Gaussian process modeling, as they encode our assumptions about the estimated mapping f. By choosing the covariance function properly, one can set the prior knowledge about the expected properties of the function, such as continuity, differentiability, smoothness, stationarity or periodicity. The covariance function can also be seen to define a similarity measure between function values $f(\mathbf{x})$ for different inputs $\mathbf{x}$. Similarity is an essential measure when doing predictions because it tells which known function values are relevant for prediction.

The range of possible covariance functions $k(\mathbf{x}, \mathbf{x}')$ is restricted by only a few simple mathematical properties. Although covariance functions are flexible, an arbitrary function $k(\mathbf{x}, \mathbf{x}')$ is not, in general, a valid covariance function. A valid covariance function is symmetric, that is, $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$. It should also be positive semidefinite, that is, for any set of input points $\{\mathbf{x}_n\}_{n=1}^N$, the resulting Gram matrix $\mathbf{K}$, whose elements are $[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, is positive semidefinite (i.e., $\mathbf{v}^{\mathrm{T}}\mathbf{K}\mathbf{v} \geq 0$ for all $\mathbf{v} \in \mathbb{R}^N$).

Next, some covariance functions are presented as examples, and they will be used in the later chapters. The considered covariance functions are isotropic, that is, they are functions of $r = d(\mathbf{x}, \mathbf{x}')$, where d is a distance measure between $\mathbf{x}$ and $\mathbf{x}'$. For instance, d could be the Euclidean distance $d(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')}$ or, in climatology, the distance between two locations on Earth measured over the surface of the sphere, as defined in Appendix A.2.

One of the most commonly used isotropic covariance functions is the squared exponential covariance function

$$k(r; \theta_1) = \exp\left(-\frac{r^2}{2\theta_1^2}\right), \tag{4.12}$$

where the parameter θ_1 is a characteristic length scale controlling the smoothness of the function. Figure 4.1a shows the covariance function with different length scales θ_1 and samples drawn from the corresponding GPs.

Periodic functions can be modeled with a periodic covariance function

$$k(r; \theta_1, \theta_2) = \exp\left(-\frac{2\sin^2(\pi r/\theta_1)}{\theta_2^2}\right), \tag{4.13}$$

where θ_1 is the length of the period and θ_2 controls the smoothness (MacKay, 1998). Figure 4.1b shows the covariance function with different smoothness θ_2 and samples from the corresponding GPs. Clearly, the periodic component does not need to be sinusoid.

A computationally interesting class of covariance functions is a family of piecewise polynomials with compact support. These covariance functions have the property that the covariance between two data points becomes exactly zero as the distance r exceeds a certain threshold. Thus, they produce sparse covariance matrices by construction, leading to computational advantages. One of the piecewise polynomial covariance functions defined in the D-dimensional real space $\mathbb{R}^D$ is

$$k(r; \theta_1) = \frac{1}{3}(1 - \hat{r})^{j+2}\left((j^2 + 4j + 3)\hat{r}^2 + (3j + 6)\hat{r} + 3\right), \tag{4.14}$$

where $\hat{r} = \min(1, r/\theta_1)$, θ_1 is the distance threshold and $j = \lfloor \frac{D}{2} \rfloor + 3$. Figure 4.1c shows the covariance function with different thresholds θ_1 and samples from the corresponding GPs.

One can derive new valid covariance functions from old ones using some basic operations. Valid operations are, for example, multiplying and adding existing covariance functions together. For instance, multiplying the squared exponential (4.12) and periodic (4.13) covariance functions results in a quasi-periodic covariance function

$$k(r; \theta_1, \theta_2, \theta_3) = \exp\left(-\frac{2\sin^2(\pi r/\theta_1)}{\theta_2^2} - \frac{r^2}{2\theta_3^2}\right), \tag{4.15}$$

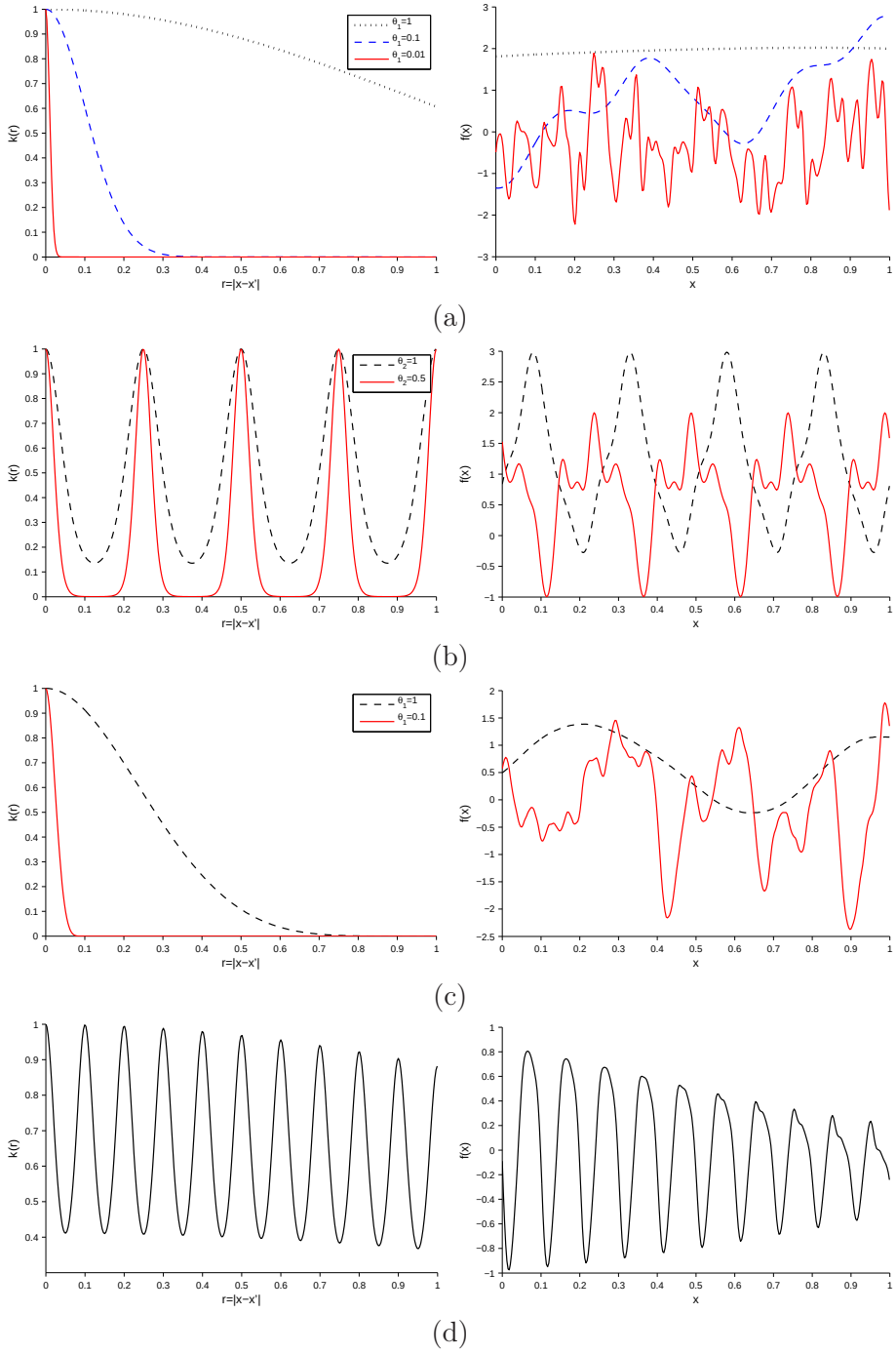
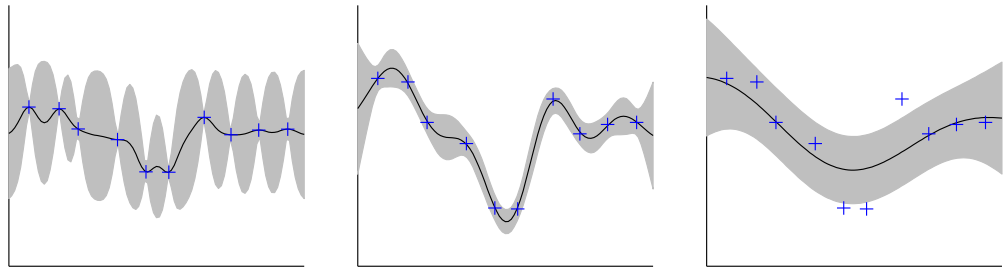


Figure 4.1: Left hand side shows the covariance function values and right hand side one-dimensional samples from the corresponding GP distribution using (a) squared exponential, (b) periodic ($\theta_1 = 0.25$), (c) piecewise polynomial, and (d) decaying periodic ($\theta_1 = 0.1$, $\theta_2 = 1.5$, $\theta_3 = 2$) covariance functions.



(a) (b) (c)

Figure 4.2: Data is generated from a GP using squared exponential covariance function, as shown by the + symbols. Three GPs are fitted to the data by using (a) shorter, (b) equal, and (c) longer length scale than was used in generating the data. The noise variance σ^2 was also set to (a) smaller, (b) equal and (c) larger values than the true noise variance. The solid line and the gray coloring show the mean and two standard deviations computed from the posterior distribution.

where θ_1 and θ_2 have similar functions as in (4.13), and θ_3 controls the decay of the periodicity. Sample functions from this covariance function have the property that they are approximately periodic: the correlation between points even in the same phase approaches zero as the distance between the points increases enough. Thus, the functions are periodic on a local scale but not necessarily on a global scale. This is illustrated in Figure 4.1d. New covariance functions are also obtained by multiplying a valid covariance function by a scalar. For instance, the squared exponential covariance function (4.12) can be scaled with an additional parameter θ_2 as

$$k(r;\theta_1,\theta_2) = \theta_2^2 \exp\left(-\frac{r^2}{2\theta_1^2}\right). \tag{4.16}$$

A more comprehensive list of examples can be found, for instance, in the book by Rasmussen and Williams (2006). In addition, Cressie and Huang (1999) discuss how to assign realistic and complex enough covariance functions to model spatio-temporal systems.

The parameters in the covariance functions are important in determining the properties of the function. Figure 4.2 illustrates the effect of the parameters by showing the posterior distribution of the underlying function given some observations by using three different values for the length scale. Clearly, each of the length scales corresponds to different interpretations of the data. In some cases, the parameters of the covariance function can be known quite accurately a priori, and then their values can be fixed. However, the parameters are usually not known a priori, and they must be learned somehow.

4.4 Learning the hyperparameters

Ideally, one would set the priors over the (hyper)parameters in the covariance function and integrate the hyperparameters out. However, they usually have such a complicated relation to the observations through the covariance function that the relevant integrals become intractable. Therefore, approximation methods, such as MCMC or maximum likelihood, are typically used (Williams and Rasmussen, 1996). The integration can be done numerically using MCMC methods, but this method can become computationally too expensive. Using gradient-based optimization methods, the hyperparameters can be set to maximum likelihood values. Although ML estimates tend to overfit, the risk of getting badly overfitted solutions is rather small if the posterior is well peaked, which is more common to the hyperparameters than to the latent function values $\mathbf{f}$ (MacKay, 1999).

The hyperparameters are optimized to maximize the marginal likelihood from which the latent function values $\mathbf{f}$ have been integrated out. The joint distribution $p(\mathbf{y}, \mathbf{f}|\mathrm{X})$ is obtained by multiplying the prior (4.4) and the likelihood (4.5), and then $\mathbf{f}$ can be integrated out, yielding

$$p(\mathbf{y}|\mathrm{X}, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{y}\,|\mathbf{0}, \mathbf{K_f} + \boldsymbol{\Sigma}\right),$$

where we have now explicitly conditioned on the hyperparameters $\boldsymbol{\theta}$, which include the parameters of the covariance function. The log marginal likelihood equals

$$\mathcal{L}(\boldsymbol{\theta}) = \log p(\mathbf{y}|\mathrm{X}, \boldsymbol{\theta}) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K_f} + \boldsymbol{\Sigma}| - \frac{1}{2}\mathbf{y}^{\mathrm{T}}(\mathbf{K_f} + \boldsymbol{\Sigma})^{-1}\mathbf{y}.$$

Note that $\mathbf{K_f}$ and $\boldsymbol{\Sigma}$ are functions of the hyperparameters. Finding maximum likelihood estimates for the hyperparameters can be implemented efficiently by utilizing the gradient

$$\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\theta_i} = \frac{1}{2}\mathbf{y}^{\mathrm{T}}(\mathbf{K_f} + \boldsymbol{\Sigma})^{-1}\frac{\partial\mathbf{K_f}}{\partial\theta_i}(\mathbf{K_f} + \boldsymbol{\Sigma})^{-1}\mathbf{y} - \frac{1}{2}\operatorname{tr}\left((\mathbf{K_f} + \boldsymbol{\Sigma})^{-1}\frac{\partial\mathbf{K_f}}{\partial\theta_i}\right),$$

where the elements of $\frac{\partial\mathbf{K_f}}{\partial\theta_i}$ are the derivatives of the covariance function.

4.5 Variational sparse approximation

One of the main issues with Gaussian processes is that they do not scale well to large problems. This issue arises because the posterior distribution requires inverting the matrix $\mathbf{K_f} + \boldsymbol{\Sigma}$ or evaluating $(\mathbf{K_f} + \boldsymbol{\Sigma})^{-1}\mathbf{y}$. The computation time of this inverse scales cubically with respect to the number of data points

N, that is, the computational complexity is $O(N^3)$. The storage requirement for the covariance matrices is $O(N^2)$, which also limits the size of solvable problems (e.g., the memory requirement for $N = 10^5$ data points using 64-bit floats is over 74 GB).

In order to reduce the computational cost and memory requirements, several sparse approximation techniques have been suggested (see, e.g., Seeger et al., 2003; Snelson and Ghahramani, 2006). The key idea is that if the data points are located densely compared to the length scale of the function, it is sufficient to know the value of the function only at some of the input points in order to recover the function at the rest of the input points accurately. These approximations are based on using a small set of $\hat{N} << N$ inducing inputs such that the corresponding function values capture the behavior of the function sufficiently well. Different sparse approximations differ in how they choose these inducing inputs and approximate the likelihood function. For instance, Seeger et al. (2003) suggested selecting inducing inputs as a subset of the inputs in the dataset and solving this combinatorial problem of finding the optimal subset by using greedy selection methods. Snelson and Ghahramani (2006) noted that the inducing inputs do not have to be selected from the dataset but they can be arbitrary points, which are then optimized by using a gradient-based optimization method.

Quiñonero-Candela and Rasmussen (2005) showed that these approximations can be interpreted as modifications of the GP prior and likelihood. Thus, the approximations can be seen as doing exact inference on an approximate model. In practice, this may increase flexibility in some unexpected way as was seen in the example by Snelson and Ghahramani (2006). A major drawback of these approaches is the lack of a distance measure between the true and the modified model to be minimized.

4.5.1 Form of approximation

Recently, Titsias (2009) applied the variational Bayesian framework to sparse approximations in GP learning. By using the VB methodology, one obtains a lower bound of the marginal likelihood, which can be used to optimize the hyperparameters and the inducing inputs. As the VB iteration minimizes the Kullback-Leibler divergence between the true and the modified model, the sparse approximation is guaranteed to get closer to the true distribution at each step until convergence.

The variational sparse approximation introduces a set of $\hat{N} << N$ auxiliary variables $\{\hat{f}_{\hat{n}}\}_{\hat{n}=1}^{\hat{N}}$ defined as $\hat{f}_{\hat{n}} = f(\hat{\mathbf{x}}_{\hat{n}})$ for inducing inputs $\hat{\mathrm{X}} = \{\hat{\mathbf{x}}_{\hat{n}}\}_{\hat{n}=1}^{\hat{N}}$. These inducing inputs can be chosen as a subset from the data inputs or by using some heuristic. In any case, they can be optimized during learning.

The goal of the sparse approximation is to find such an approximate pos-

terior distribution that the auxiliary variables $\hat{\mathbf{f}}$ are used to summarize the data. The true joint posterior of the auxiliary variables and the original latent function values $\mathbf{f}$ can be written as

$$p(\mathbf{f}, \hat{\mathbf{f}}|\mathbf{y}) = p(\mathbf{f}|\hat{\mathbf{f}}, \mathbf{y})p(\hat{\mathbf{f}}|\mathbf{y}),$$

where $\hat{\mathbf{f}} = \begin{bmatrix}\hat{f}_1 & \dots & \hat{f}_{\hat{N}}\end{bmatrix}^{\mathrm{T}}$ contains the auxiliary variables. To keep the notation compact, we have omitted the conditioning on the inputs X and $\hat{\mathrm{X}}$. If the inducing inputs are located sufficiently densely, the auxiliary variables $\hat{\mathbf{f}}$ summarize the information about the function well, and it holds that $p(\mathbf{f}|\hat{\mathbf{f}}, \mathbf{y}) \approx p(\mathbf{f}|\hat{\mathbf{f}})$, that is, the data does not give any additional information compared to the auxiliary variables when recovering the latent function. This suggests a convenient form of the approximate posterior:

$$q(\mathbf{f}, \hat{\mathbf{f}}) = p(\mathbf{f}|\hat{\mathbf{f}})q(\hat{\mathbf{f}}). \tag{4.17}$$

The factor $p(\mathbf{f}|\hat{\mathbf{f}})$ can easily be computed from the GP prior by using the conditional Gaussian distribution (A.6). The factor $q(\hat{\mathbf{f}})$ is found by free-form maximization of the variational lower bound of the marginal likelihood, as discussed generally in Section 2.3.2.

4.5.2 Approximate posterior distribution

Starting from the definition in (2.4), the lower bound of the log marginal likelihood can be written as

$$\begin{aligned}
\mathcal{L}\left(q(\mathbf{f}, \hat{\mathbf{f}}), \boldsymbol{\theta}, \hat{\mathrm{X}}\right) &= \iint q(\mathbf{f}, \hat{\mathbf{f}}) \log \frac{p(\mathbf{y}, \mathbf{f}, \hat{\mathbf{f}})}{q(\mathbf{f}, \hat{\mathbf{f}})} d\mathbf{f} d\hat{\mathbf{f}} \\
&= \iint p(\mathbf{f}|\hat{\mathbf{f}})q(\hat{\mathbf{f}}) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\hat{\mathbf{f}})p(\hat{\mathbf{f}})}{p(\mathbf{f}|\hat{\mathbf{f}})q(\hat{\mathbf{f}})} d\mathbf{f} d\hat{\mathbf{f}} \\
&= \iint p(\mathbf{f}|\hat{\mathbf{f}})q(\hat{\mathbf{f}}) \log \frac{p(\mathbf{y}|\mathbf{f})p(\hat{\mathbf{f}})}{q(\hat{\mathbf{f}})} d\mathbf{f} d\hat{\mathbf{f}} \\
&= \int q(\hat{\mathbf{f}}) \left[\int p(\mathbf{f}|\hat{\mathbf{f}}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} + \log \frac{p(\hat{\mathbf{f}})}{q(\hat{\mathbf{f}})} \right] d\hat{\mathbf{f}} \\
&= \int q(\hat{\mathbf{f}}) \left[\log \tilde{p}(\mathbf{y}|\hat{\mathbf{f}}) + \log \frac{p(\hat{\mathbf{f}})}{q(\hat{\mathbf{f}})} \right] d\hat{\mathbf{f}} \\
&= \int q(\hat{\mathbf{f}}) \log \frac{\tilde{p}(\mathbf{y}|\hat{\mathbf{f}})p(\hat{\mathbf{f}})}{q(\hat{\mathbf{f}})} d\hat{\mathbf{f}},
\end{aligned} \tag{4.18}$$

where the second line is obtained by expressing the factors of the joint distributions $q(\mathbf{f}, \hat{\mathbf{f}})$ and $p(\mathbf{y}, \mathbf{f}, \hat{\mathbf{f}})$, the third line by canceling out $p(\mathbf{f}|\hat{\mathbf{f}})$ inside the logarithm, the fourth line by reorganizing the terms, the fifth line by denoting

$$\begin{aligned}\log \tilde{p}(\mathbf{y}|\hat{\mathbf{f}}) &= \int p(\mathbf{f}|\hat{\mathbf{f}}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} \\ &= \log \mathcal{N}\left(\mathbf{y} \left| \mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}} \mathbf{K}_{\hat{\mathbf{f}}}^{-1} \hat{\mathbf{f}}, \boldsymbol{\Sigma} \right.\right) - \frac{1}{2} \operatorname{tr}\left(\operatorname{cov}\left(\mathbf{f} \left| \hat{\mathbf{f}} \right.\right) \boldsymbol{\Sigma}^{-1}\right), \end{aligned} \tag{4.19}$$

and the final line by reorganizing the terms. We have also used the following notation:

$$\operatorname{cov}\left(\mathbf{f} \left| \hat{\mathbf{f}} \right.\right) = \mathbf{K}_{\mathbf{f}} - \mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}} \mathbf{K}_{\mathbf{f}}^{-1} \mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}}, \tag{4.20}$$

as defined for the conditional Gaussian distribution in (A.6). Note that (4.20) does not depend on the function values $\hat{\mathbf{f}}$ but on the inducing inputs $\hat{\mathbf{X}}$ and the hyperparameters $\boldsymbol{\theta}$.

The optimal $q(\hat{\mathbf{f}})$ can be found by interpreting (4.18) as the Kullback-Leibler divergence between $q(\hat{\mathbf{f}})$ and $\tilde{p}(\mathbf{y}|\hat{\mathbf{f}})p(\hat{\mathbf{f}})$. It follows that the optimal $q(\hat{\mathbf{f}})$ is proportional to $\tilde{p}(\mathbf{y}|\hat{\mathbf{f}})p(\hat{\mathbf{f}})$, and can thus be seen as the posterior distribution of a model with a likelihood function $\tilde{p}(\mathbf{y}|\hat{\mathbf{f}})$ and a prior $p(\mathbf{f})$. This posterior has the form

$$q(\hat{\mathbf{f}}) = \mathcal{N}\left(\hat{\mathbf{f}} \left| \mathbf{V} \mathbf{K}_{\hat{\mathbf{f}}}^{-1} \mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}} \boldsymbol{\Sigma}^{-1} \mathbf{y}, \mathbf{V}_{\hat{\mathbf{f}}} \right.\right), \tag{4.21}$$

where we have denoted

$$\mathbf{V}_{\hat{\mathbf{f}}} = \left(\mathbf{K}_{\hat{\mathbf{f}}}^{-1} + \mathbf{K}_{\hat{\mathbf{f}}}^{-1} \mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}} \boldsymbol{\Sigma}^{-1} \mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}} \mathbf{K}_{\hat{\mathbf{f}}}^{-1}\right)^{-1},$$

which is similar to the posterior (4.6) in regular GP with the difference that the output space is projected to the lower dimensional space of the auxiliary variables $\hat{\mathbf{f}}$. Note that the approximate posterior $q(\hat{\mathbf{f}})$ is different from the true marginal posterior $p(\hat{\mathbf{f}}|\mathbf{y})$ in (4.9).

The sparse approximate posterior distribution (4.21) is computationally much more feasible than the full posterior distribution (4.6), because the matrix inversion is done to a matrix of size $\hat{N} \times \hat{N}$ instead of $N \times N$. Assuming that one can efficiently evaluate the term $\boldsymbol{\Sigma}^{-1} \mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}}$ because of a very sparse structure of $\boldsymbol{\Sigma}$, for example, (block-)diagonality, the main cost comes from the term $\mathbf{K}_{\hat{\mathbf{f}}}^{-1} \mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}}$. Thus, the time complexity of the sparse approximation is $O(\hat{N}^2 N)$. In addition, the approximate posterior (4.21) reduces the memory requirement to $O(\hat{N}N)$ as it does not require the evaluation of the large covariance matrix $\mathbf{K}_{\mathbf{f}}$.

The posterior predictive distribution is evaluated by using the posterior of the auxiliary variables. The function values $\tilde{\mathbf{f}}$ for any new inputs $\tilde{\mathrm{X}} = \{\tilde{\mathbf{x}}_{\tilde{n}}\}_{\tilde{n}=1}^{\tilde{N}}$ are modeled with the posterior distribution $q(\tilde{\mathbf{f}}) = \int p(\tilde{\mathbf{f}}|\hat{\mathbf{f}})q(\hat{\mathbf{f}})d\hat{\mathbf{f}}$, which equals

$$q(\tilde{\mathbf{f}}) = \mathcal{N}\left(\tilde{\mathbf{f}} \left| \mathbf{K}_{\tilde{\mathbf{f}},\hat{\mathbf{f}}}\boldsymbol{\Lambda}^{-1}\mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}}\boldsymbol{\Sigma}^{-1}\mathbf{y}, \mathbf{K}_{\tilde{\mathbf{f}}} - \mathbf{K}_{\tilde{\mathbf{f}},\hat{\mathbf{f}}}\left(\mathbf{K}_{\hat{\mathbf{f}}}^{-1} - \boldsymbol{\Lambda}^{-1}\right)\mathbf{K}_{\hat{\mathbf{f}},\tilde{\mathbf{f}}}\right.\right),$$

where we have denoted

$$\boldsymbol{\Lambda} = \mathbf{K}_{\hat{\mathbf{f}}} + \mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}}\boldsymbol{\Sigma}^{-1}\mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}}. \tag{4.22}$$

This distribution is derived in Appendix B.2. Evaluating the full covariance matrix of the predictive distribution can be computationally expensive as it includes the term $\mathbf{K}_{\tilde{\mathbf{f}}}$. Thus, instead of evaluating the full covariance, it is often sufficient to only evaluate the variances, that is, the diagonal elements of the covariance matrix.

4.5.3 Variational lower bound

Let us derive a lower bound of the log marginal likelihood for optimizing the hyperparameters and the inducing inputs. To begin with, we insert the normalized approximate posterior distribution $q(\hat{\mathbf{f}}) = \tilde{p}(\mathbf{y}|\hat{\mathbf{f}})p(\hat{\mathbf{f}})/\int \tilde{p}(\mathbf{y}|\hat{\mathbf{f}})p(\hat{\mathbf{f}})d\hat{\mathbf{f}}$ to the lower bound (4.18), yielding

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \hat{\mathrm{X}}) &= \log \int \tilde{p}(\mathbf{y}|\hat{\mathbf{f}})p(\hat{\mathbf{f}})d\hat{\mathbf{f}} \\ &= \log \mathcal{N}\left(\mathbf{y} \left| \mathbf{0}, \boldsymbol{\Sigma} + \mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}}\mathbf{K}_{\hat{\mathbf{f}}}^{-1}\mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}}\right.\right) - \frac{1}{2}\operatorname{tr}\left(\operatorname{cov}\left(\mathbf{f}|\hat{\mathbf{f}}\right)\boldsymbol{\Sigma}^{-1}\right).\end{aligned} \tag{4.23}$$

The first term encourages the approximation to fit well to the data. The second term penalizes for the conditional covariance of $\mathbf{f}$ given $\hat{\mathbf{f}}$, that is, the uncertainty in predicting $\mathbf{f}$ from the auxiliary variables $\hat{\mathbf{f}}$. Although the conditional covariance $\operatorname{cov}\left(\mathbf{f}\left|\hat{\mathbf{f}}\right.\right)$ includes the full covariance matrix $\mathbf{K}_{\mathbf{f}}$, as shown in (4.20), this computationally heavy evaluation can be avoided if the inverse noise covariance $\boldsymbol{\Sigma}^{-1}$ is sparse. This follows from the fact that in order to evaluate the trace term in (4.23), one needs to evaluate only those elements of $\operatorname{cov}\left(\mathbf{f}\left|\hat{\mathbf{f}}\right.\right)$ that correspond to the nonzero elements of $\boldsymbol{\Sigma}^{-1}$.

Now, the hyperparameters $\boldsymbol{\theta}$ in the covariance function and the inducing inputs $\hat{\mathrm{X}}$ can be chosen by maximizing the lower bound (4.23). The optimization of the hyperparameters can be implemented efficiently by exploiting the

gradient

$$\begin{aligned}\frac{\partial \mathcal{L}(\boldsymbol{\theta}, \hat{\mathrm{X}})}{\partial \theta_i} =& \frac{1}{2} \operatorname{tr}\left[\left(\mathbf{K}_{\hat{\mathbf{f}}}^{-1} - \boldsymbol{\Lambda}^{-1}\right) \frac{\partial \mathbf{K}_{\hat{\mathbf{f}}}}{\partial \theta_i}\right] - \operatorname{tr}\left(\boldsymbol{\Sigma}^{-1} \mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}} \boldsymbol{\Lambda}^{-1} \frac{\partial \mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}}}{\partial \theta_i}\right) - \\ & \frac{1}{2} \boldsymbol{\nu}^{\mathrm{T}} \frac{\partial \mathbf{K}_{\hat{\mathbf{f}}}}{\partial \theta_i} \boldsymbol{\nu} + \boldsymbol{\nu}^{\mathrm{T}} \frac{\partial \mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}}}{\partial \theta_i} \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}} \boldsymbol{\nu}) + \\ & \operatorname{tr}\left(\frac{\partial \mathbf{K}_{\mathbf{f}}}{\partial \theta_i} \boldsymbol{\Sigma}^{-1}\right) + \operatorname{tr}\left(\frac{\partial \mathbf{K}_{\hat{\mathbf{f}}}}{\partial \theta_i} \mathbf{K}_{\hat{\mathbf{f}}}^{-1} \mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}} \boldsymbol{\Sigma}^{-1} \mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}} \mathbf{K}_{\hat{\mathbf{f}}}^{-1}\right) - \\ & 2 \operatorname{tr}\left(\frac{\partial \mathbf{K}_{\mathbf{f},\hat{\mathbf{f}}}}{\partial \theta_i} \mathbf{K}_{\hat{\mathbf{f}}}^{-1} \mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}} \boldsymbol{\Sigma}^{-1}\right),\end{aligned}$$

where $\boldsymbol{\Lambda}$ is defined in (4.22) and $\boldsymbol{\nu} = \boldsymbol{\Lambda}^{-1} \mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}} \boldsymbol{\Sigma}^{-1} \mathbf{y}$. The derivation of this gradient is shown in Appendix B.2.

Note that the optimization of the hyperparameters $\boldsymbol{\theta}$ and the inducing inputs $\hat{\mathrm{X}}$ does not depend on the explicit form of the approximate posterior distribution $q(\hat{\mathbf{f}})$. On the other hand, $q(\hat{\mathbf{f}})$ in (4.21) is conditional on the hyperparameters and the inducing inputs through the covariance matrices $\mathbf{K}_{\hat{\mathbf{f}}}$ and $\mathbf{K}_{\hat{\mathbf{f}},\mathbf{f}}$. Therefore, one should first optimize $\boldsymbol{\theta}$ and $\hat{\mathrm{X}}$, and then evaluate the approximate posterior $q(\hat{\mathbf{f}})$.

4.6 Conclusions

This chapter presented Gaussian processes as a flexible tool for nonlinear regression problems. The infinite-dimensional Gaussian process distribution was seen as a consistent way of setting finite-dimensional marginal Gaussian distributions. We presented some important covariance functions and illustrated them by showing samples from the corresponding GPs. Because the values of the hyperparameters are important, they can be learned by using gradient-based optimization methods. We also explained in detail the state-of-the-art variational sparse approximation to reduce the computational cost.

Chapter 5

Gaussian-process factor analysis

This chapter introduces a novel method for spatio-temporal modeling and exploratory data analysis (Luttinen and Ilin, 2009). The model is defined in Section 5.1 as an extension to factor analysis in which the loading matrix $\mathbf{A}$ and the states $\mathbf{S}$ are given Gaussian process priors. Because the resulting posterior distribution is analytically intractable, we apply the variational Bayesian methodology to find an approximate posterior distribution. Section 5.2 derives the approximation focusing on the states $\mathbf{S}$. As the Gaussian process priors raise the computational cost extremely high for large datasets, Section 5.3 presents sparse approximations to reduce the computational burden. Sections 5.2 and 5.3 also explain how to further reduce the computational cost by factorizing with respect to the components and how to find maximum likelihood estimates for the hyperparameters. Section 5.4 explains how to learn the other variables in the model. Because the main problem with the model is the high computational cost, Section 5.5 discusses some implementation issues. Section 5.6 briefly discusses other closely related models.

5.1 Model

Let us consider spatio-temporal data which consists of observations $y_{mn} = y(l_m, t_n)$ at spatial locations $\{l_m\}_{m=1}^M$ at time instances $\{t_n\}_{n=1}^N$. We model the data using the factor analysis model defined in (3.3):

$$y_{mn} = \mathbf{a}_{m:}^{\mathrm{T}} \mathbf{s}_{:n} + \text{noise}, \quad m = 1, \ldots, M, \quad n = 1, \ldots, N \tag{5.1}$$

where $\mathbf{a}_{m:}^{\mathrm{T}}$ is the m-th row of the $M \times D$ loading matrix $\mathbf{A}$ and $\mathbf{s}_{:n}$ is the n-th column of the $D \times N$ matrix $\mathbf{S}$ of factors. For simplicity, we have discarded the bias term $\boldsymbol{\mu}$ but it is straightforward to add it or model the bias by fixing one row of $\mathbf{S}$ to ones. We can also summarize the entire dataset in a matrix

form

$$\mathbf{Y} = \sum_{d=1}^{D} \mathbf{a}_{:d}\mathbf{s}_{d:}^{\mathrm{T}} + \text{noise}, \tag{5.2}$$

where $\mathbf{Y}$ is a $M \times N$ matrix containing the elements y_{mn} on the m-th row and n-th column, $\mathbf{a}_{:d}$ is the d-th column of $\mathbf{A}$ and $\mathbf{s}_{d:}^{\mathrm{T}}$ is the d-th row of $\mathbf{S}$. The vectors $\mathbf{a}_{:d}$ and $\mathbf{s}_{d:}$ can be seen as spatial and temporal patterns, respectively.

In order to model complex spatial and temporal structure, the prior distributions of the patterns $\mathbf{a}_{:d}$ and $\mathbf{s}_{d:}$ are set to Gaussian processes:

$$p(\mathbf{A}) = \prod_{d=1}^{D} \mathcal{N}\left(\mathbf{a}_{:d} \,|\mathbf{0}, \mathbf{K}_{\mathbf{a}_{:d}}\right) = \mathcal{N}\left(\mathbf{a}_{:} \,|\mathbf{0}, \mathbf{K}_{\mathbf{a}_{:}}\right), \tag{5.3}$$

$$p(\mathbf{S}) = \prod_{d=1}^{D} \mathcal{N}\left(\mathbf{s}_{d:} \,|\mathbf{0}, \mathbf{K}_{\mathbf{s}_{d:}}\right) = \mathcal{N}\left(\mathbf{s}_{:} \,|\mathbf{0}, \mathbf{K}_{\mathbf{s}_{:}}\right), \tag{5.4}$$

where $\mathbf{a}_{:}$ and $\mathbf{s}_{:}$ denote long vectors formed from all elements of $\mathbf{A}$ and $\mathbf{S}$, respectively, and the covariance matrices are defined as

$$\left[\mathbf{K}_{\mathbf{a}_{:d}}\right]_{ij} = k_{a_d}(l_i, l_j; \boldsymbol{\phi}_d),$$
$$\left[\mathbf{K}_{\mathbf{s}_{d:}}\right]_{ij} = k_{s_d}(t_i, t_j; \boldsymbol{\theta}_d),$$

where $\boldsymbol{\phi}_d$ and $\boldsymbol{\theta}_d$ are the hyperparameters of the covariance functions. The ordering of the elements in $\mathbf{a}_{:}$ and $\mathbf{s}_{:}$ can be arbitrary as long as they are consistent with the ordering of the rows and columns of the covariance matrices $\mathbf{K}_{\mathbf{a}_{:}}$ and $\mathbf{K}_{\mathbf{s}_{:}}$. If the elements are ordered component-wise (i.e., column-wise for $\mathbf{A}$ and row-wise for $\mathbf{S}$), the resulting covariance matrices are block-diagonal with blocks $\mathbf{K}_{\mathbf{a}_{d:}}$ and $\mathbf{K}_{\mathbf{s}_{d:}}$ on the diagonal, because the components are independent a priori.

The spatial and temporal patterns can be interpreted as functions $a_d(l)$ and $s_d(t)$, which have the following prior distributions:

$$a_d(l) \sim \mathcal{GP}\left(0, k_{a_d}(l, l'; \boldsymbol{\phi}_d)\right),$$
$$s_d(t) \sim \mathcal{GP}\left(0, k_{s_d}(t, t'; \boldsymbol{\theta}_d)\right).$$

This functional viewpoint makes it more clear that the inputs $\{l_m\}_{m=1}^{M}$ and $\{t_n\}_{n=1}^{N}$ do not need to form any regular grid. This viewpoint also emphasizes that we are modeling a spatio-temporal function $y(l, t)$ as

$$y(l, t) = \sum_{d=1}^{D} a_d(l)s_d(t) + \text{noise}. \tag{5.5}$$

The value of this function can be predicted at arbitrary locations at arbitrary times.

The noise term in (5.1), (5.2) and (5.5) can be modeled simply as Gaussian noise, resulting in a likelihood function

$$p(\mathbf{Y}|\mathbf{A},\mathbf{S},\boldsymbol{\tau}) = \prod_{m=1}^{M}\prod_{n=1}^{N}\mathcal{N}\left(y_{mn}\left|\mathbf{a}_{m:}^{\mathrm{T}}\mathbf{s}_{:n},\tau_{mn}^{-1}\right.\right). \tag{5.6}$$

Instead of using spatio-temporally varying noise level, the noise can be assumed to vary only temporally ($\tau_{mn} = \tau_n$) or spatially ($\tau_{mn} = \tau_m$), as in standard factor analysis. However, if the model used independently varying noise levels, it could no longer make predictions at a new location because it has no information about the noise level at that location. This issue could be solved by using a generative hierarchical model for τ_{mn}. However, the dataset may contain very little evidence for spatially or temporally varying noise, and modeling such variability would complicate the modeling quite remarkably. Therefore, isotropic noise ($\tau_{mn} = \tau$) can often be a sufficient compromise in practice. For areal data, the noise levels may be weighted by the size of the area.

The conjugate prior for τ_{mn} is the gamma distribution:

$$p(\tau_{mn}) = \mathcal{G}\left(\tau_{mn}\left|\alpha_\tau,\beta_\tau\right.\right). \tag{5.7}$$

In order to obtain a broad prior, α_τ and β_τ may be set to small positive values, for instance, 10^{-3}. We will refer to the model defined in (5.1)–(5.7) as Gaussian-process factor analysis (GPFA).

The GPFA model has a few advantages compared to the regular GP regression over the spatio-temporal domain: (1) GPFA models spatial and temporal functions separately by using D independent Gaussian processes for both domains. If $D << M$ and $D << N$, the savings may be significant compared to the computational cost $O(M^3N^3)$ of regular GP regression. (2) GPFA extracts spatial and temporal features, which are easily interpreted, and making the selection of the priors for these spatial and temporal features intuitive. (3) The factor analysis modeling assumption might be quite reasonable, because the variability of a multidimensional process can often be captured in a low-dimensional representation.

The model has two major issues which should be addressed: The posterior $p(\mathbf{A},\mathbf{S},\boldsymbol{\tau}|\mathbf{Y})$ is intractable, and the computational load for dealing with GPs becomes too high for large datasets. The variational Bayesian framework is used to cope with these difficulties by finding a tractable posterior approximation and allowing sparse approximations for the GPs.

5.2 Variational full approximation for S

5.2.1 Approximate posterior distribution

In order to find a tractable posterior approximation, we apply the variational Bayesian framework. The true posterior is approximated by a factorized distribution as

$$p(\mathbf{A}, \mathbf{S}, \boldsymbol{\tau}|\mathbf{Y}) \approx q(\mathbf{A}, \mathbf{S}, \boldsymbol{\tau}) \equiv q(\mathbf{A})q(\mathbf{S})q(\boldsymbol{\tau}).$$

The optimal approximation $q(\mathbf{A}, \mathbf{S}, \boldsymbol{\tau})$ is found by minimizing the Kullback-Leibler divergence to the true posterior, which is equivalent to maximizing the lower bound of the log marginal likelihood.

In order to maximize with respect to $q(\mathbf{S})$, the lower bound can be written as

$$\begin{aligned}
\mathcal{L}\left(q(\mathbf{S}), \boldsymbol{\Theta}\right) &= \int q(\mathbf{A})q(\mathbf{S})q(\boldsymbol{\tau}) \log \frac{p(\mathbf{Y}|\mathbf{A}, \mathbf{S}, \boldsymbol{\tau})p(\mathbf{A})p(\mathbf{S})p(\boldsymbol{\tau})}{q(\mathbf{A})q(\mathbf{S})q(\boldsymbol{\tau})} d\mathbf{A} d\mathbf{S} d\boldsymbol{\tau} \\
&= \int q(\mathbf{S}) \log \frac{\tilde{p}(\mathbf{Y}|\mathbf{S})p(\mathbf{S})}{q(\mathbf{S})} d\mathbf{S}, \qquad (5.8) \\
&= \int q(\mathbf{S}) \left[\log \tilde{p}(\mathbf{Y}|\mathbf{S}) + \log \frac{p(\mathbf{S})}{q(\mathbf{S})}\right] d\mathbf{S},
\end{aligned}$$

where

$$\begin{aligned}
\log \tilde{p}(\mathbf{Y}|\mathbf{S}) &= \int q(\mathbf{A})q(\boldsymbol{\tau}) \log \frac{p(\mathbf{Y}|\mathbf{A}, \mathbf{S}, \boldsymbol{\tau})p(\mathbf{A})p(\boldsymbol{\tau})}{q(\mathbf{A})q(\boldsymbol{\tau})} d\mathbf{A} d\boldsymbol{\tau} \\
&= \langle \log p(\mathbf{Y}|\mathbf{A}, \mathbf{S}, \boldsymbol{\tau}) \rangle + \text{const} \\
&= \log \mathcal{N}\left(\mathbf{U}^{-1}\mathbf{z}_{:} \left| \mathbf{s}_{:}, \mathbf{U}^{-1}\right.\right) + \text{const}. \qquad (5.9)
\end{aligned}$$

The notation const refers to the terms that are constant with respect to $\mathbf{S}$, and $\mathbf{z}_{:}$ is a $DN \times 1$ vector formed by concatenating vectors

$$\mathbf{z}_{:n} = \sum_{m \in \mathcal{O}_{:n}} \langle \tau_{mn} \rangle \langle \mathbf{a}_{m:} \rangle y_{mn}, \qquad n = 1, \ldots, N. \qquad (5.10)$$

Matrix $\mathbf{U}$ in (5.9) is a $DN \times DN$ block-diagonal matrix with the following $D \times D$ matrices on the diagonal:

$$\mathbf{U}_n = \sum_{m \in \mathcal{O}_{:n}} \langle \tau_{mn} \rangle \left\langle \mathbf{a}_{m:} \mathbf{a}_{m:}^{\mathrm{T}} \right\rangle, \qquad n = 1, \ldots, N. \qquad (5.11)$$

The summations in (5.10) and (5.11) are over sets $\mathcal{O}_{:n}$ of indices m for which y_{mn} is observed (i.e., $\mathcal{O}_{:n} = \{m|y_{mn} \text{ is observed}\}$), and $\langle \cdot \rangle$ is the expectation

over the approximate posterior distribution q. Note that if the elements in $\mathbf{s}_{:}$ are ordered component-wise making the covariance matrix $\mathbf{K}_{\mathbf{s}_{:}}$ block-diagonal, the elements of $\mathbf{U}$ are scattered on the $(d-1)N$-th diagonals, where $d = 1, \ldots, D$. Thus, these matrices have overlapping nonzero elements only on the main diagonal. In addition, the ordering of the elements in $\mathbf{z}_{:}$ should also correspond to the ordering in $\mathbf{s}_{:}$.

The optimal $q(\mathbf{S})$ can be found by interpreting (5.8) as the Kullback-Leibler divergence between $q(\mathbf{S})$ and $\tilde{p}(\mathbf{Y}|\mathbf{S})p(\mathbf{S})$. Thus, the optimal $q(\mathbf{S})$ is proportional to $\tilde{p}(\mathbf{Y}|\mathbf{S})p(\mathbf{S})$ and can be seen as the posterior distribution in a model with a likelihood function $\tilde{p}(\mathbf{Y}|\mathbf{S})$ and a prior $p(\mathbf{S})$. Applying the formula of the posterior Gaussian distribution (2.3) yields the optimal $q(\mathbf{S})$ as

$$q(\mathbf{S}) = \mathcal{N}\left(\mathbf{s}_{:} \,\middle|\, \left(\mathbf{K}_{\mathbf{s}_{:}}^{-1} + \mathbf{U}\right)^{-1} \mathbf{z}_{:}, \left(\mathbf{K}_{\mathbf{s}_{:}}^{-1} + \mathbf{U}\right)^{-1}\right). \tag{5.12}$$

Note that the form of this approximate posterior is similar to the posterior (4.6) in standard GP regression: if one interprets $\mathbf{U}^{-1}\mathbf{z}_{:}$ as noisy observations, and $\mathbf{U}^{-1}$ as the noise covariance matrix in the likelihood (4.5), then $q(\mathbf{S})$ equals the posterior distribution (4.6) of the latent functions values. Thus, each update of $q(\mathbf{S})$ can be seen as applying standard GP regression to projected observations, where $\mathbf{A}$ and $\boldsymbol{\tau}$ define the projection.

5.2.2 Component-wise factorization

In practice, one may need to further factorize the posterior approximation in order to reduce the computational burden. This can be done in two ways: by neglecting the posterior correlations between different components $\mathbf{s}_{d:}$ (and between spatial patterns $\mathbf{a}_{:d}$, respectively) or by neglecting the posterior correlations between different time instances $\mathbf{s}_{:n}$ (and between spatial locations $\mathbf{a}_{m:}$, respectively). The first way is computationally more expensive but captures much stronger posterior correlations which arise with Gaussian processes. Therefore, we apply the former posterior approximation:

$$q(\mathbf{S}) = \prod_{d=1}^{D} q(\mathbf{s}_{d:}).$$

The approximate posterior distribution $q(\mathbf{s}_{d:})$ can be derived similarly as the full approximate posterior distribution $q(\mathbf{S})$ in (5.12). The optimal $q(\mathbf{s}_{d:})$ is proportional to $\tilde{p}(\mathbf{Y}|\mathbf{s}_{d:})p(\mathbf{s}_{d:})$, where the pseudo-likelihood term $\tilde{p}(\mathbf{Y}|\mathbf{s}_{d:})$ is now defined as

$$\tilde{p}(\mathbf{Y}|\mathbf{s}_{d:}) \propto \mathcal{N}\left(\mathbf{U}_d^{-1}\mathbf{z}_d \,\middle|\, \mathbf{s}_{d:}, \mathbf{U}_d^{-1}\right),$$

where $\mathbf{z}_d$ is an $N \times 1$ vector and $\mathbf{U}_d$ is an $N \times N$ diagonal matrix defined as

$$[\mathbf{z}_d]_n = \sum_{m \in \mathcal{O}_{:n}} \langle \tau \rangle \langle a_{md} \rangle \left(y_{mn} - \sum_{j \neq d} \langle a_{mj} \rangle \langle x_{jn} \rangle \right), \tag{5.13}$$

$$[\mathbf{U}_d]_{nn} = \sum_{m \in \mathcal{O}_{:n}} \langle \tau \rangle \left\langle a_{md}^2 \right\rangle. \tag{5.14}$$

Applying the formula of the posterior Gaussian distribution (2.3) yields

$$q(\mathbf{s}_{d:}) = \mathcal{N}\left(\mathbf{s}_{d:} \middle| \left(\mathbf{K}_{\mathbf{s}_{d:}}^{-1} + \mathbf{U}_d \right)^{-1} \mathbf{z}_d, \left(\mathbf{K}_{\mathbf{s}_{d:}}^{-1} + \mathbf{U}_d \right)^{-1} \right), \tag{5.15}$$

The main difference to (5.12) is that each component is fitted to the residuals of the reconstruction based on the rest of the components.

Although the component-wise factorization ignores even more posterior uncertainty than the full posterior (5.12), it may provide a meaningful representation of data because the model is biased in favor of solutions with dynamically and spatially decoupled components. When the factors are modeled using rather general covariance functions, the proposed method is somewhat related to the blind source separation techniques using time structure, discussed in Section 3.4. The advantage here is that the method can handle more sophisticated temporal correlations and it is easily applicable to incomplete data. In addition, one can use the method in semi-blind settings when the prior knowledge is used to select a proper type of covariance functions. This would enable extraction of components with specific types of temporal or spatial features.

5.2.3 Learning the hyperparameters

The hyperparameters $\boldsymbol{\theta}_d$ can be point estimated by maximizing the lower bound of the log marginal likelihood. By inserting the normalized approximate posterior distribution $q(\mathbf{S}) = \tilde{p}(\mathbf{Y}|\mathbf{S})p(\mathbf{S}) / \int \tilde{p}(\mathbf{Y}|\mathbf{S})p(\mathbf{S})d\mathbf{S}$ to the lower bound (5.8), it follows that the lower bound of the marginal likelihood equals

$$\begin{aligned} \mathcal{L}(\boldsymbol{\Theta}) &= \log \int \tilde{p}(\mathbf{Y}|\mathbf{S})p(\mathbf{S})d\mathbf{S}, \\ &= \log \mathcal{N}\left(\mathbf{U}^{-1}\mathbf{z} \middle| \mathbf{0}, \mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_:} \right) + \text{const.} \end{aligned} \tag{5.16}$$

Similarly, the lower bound for the approximation using component-wise factorization equals

$$\mathcal{L}(\boldsymbol{\theta}_d) = \log \mathcal{N}\left(\mathbf{U}_d^{-1}\mathbf{z}_d \middle| \mathbf{0}, \mathbf{U}_d^{-1} + \mathbf{K}_{\mathbf{s}_{d:}} \right) + \text{const} \tag{5.17}$$

These bounds can be used to optimize the hyperparameters $\boldsymbol{\theta}_d$ of the covariance functions $k_{s_d}(t, t'; \boldsymbol{\theta}_d)$. Note that $q(\mathbf{S})$ does not need to be explicitly

evaluated in order to optimize the hyperparameters. Thus, one should first optimize the hyperparameters, and then evaluate the approximate posterior $q(\mathbf{S})$ while keeping the hyperparameters fixed. Evaluation of the lower bound and its gradients is discussed in Appendix B.1.

5.3 Variational sparse approximation for S

5.3.1 Approximate posterior distribution

One of the main issues with Gaussian processes is the high computational cost with respect to the number of observations. Although the variational learning of the GPFA model works only in either the spatial or temporal domain at a time, the size of the data may still be too large in practice, as the time complexity $O(D^3N^3 + D^3M^3)$ scales cubically with respect to N and M. This computational load can be reduced by applying the variational sparse approximations to the GP components, as discussed in Section 4.5 for the standard GP regression.

The sparse approximation introduces auxiliary variables $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ which contain the values of the latent functions $a_d(l)$ and $s_d(t)$ for a small number of inducing inputs $\hat{\mathrm{L}} = \{\hat{l}_{d\hat{m}}\}_{\hat{m}=1}^{\hat{M}_d}$ and $\hat{\mathrm{T}} = \{\hat{t}_{d\hat{n}}\}_{\hat{n}=1}^{\hat{N}_d}$. Although the numbers $\hat{M}_d << M$ and $\hat{N}_d << N$ of inducing inputs for each of the D components may be different, we have denoted the collection of the auxiliary variables

$$\hat{\mathbf{a}}_{:d} = \left[a_d(\hat{l}_{d1}) \dots a_d(\hat{l}_{d\hat{M}_d})\right]^{\mathrm{T}} \quad \text{and} \quad \hat{\mathbf{s}}_{d:} = \left[s_d(\hat{t}_{d1}) \dots s_d(\hat{t}_{d\hat{N}_d})\right]^{\mathrm{T}}$$

as matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ because this imprecise notation is less cluttered. For the same reason, we will not explicitly show the conditioning on the inducing inputs $\hat{\mathrm{L}}$ and $\hat{\mathrm{T}}$.

Assuming that the auxiliary variables $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ summarize the data well, it holds that $p(\mathbf{A}, \mathbf{S}|\hat{\mathbf{A}}, \hat{\mathbf{S}}, \mathbf{Y}) \approx p(\mathbf{A}, \mathbf{S}|\hat{\mathbf{A}}, \hat{\mathbf{S}})$. This motivates an approximate distribution of the form

$$q(\mathbf{A}, \mathbf{S}, \hat{\mathbf{A}}, \hat{\mathbf{S}}) = p(\mathbf{A}|\hat{\mathbf{A}})p(\mathbf{S}|\hat{\mathbf{S}})q(\hat{\mathbf{A}})q(\hat{\mathbf{S}}),$$

where $p(\mathbf{A}|\hat{\mathbf{A}})$, $p(\mathbf{S}|\hat{\mathbf{S}})$ can be easily computed from the GP priors by applying the conditional Gaussian distribution (A.6).

Optimal $q(\hat{\mathbf{S}})$ can be found by maximizing the variational lower bound of the log marginal likelihood. In order to optimize $q(\hat{\mathbf{S}})$, we take similar steps as in (4.18), and write the log marginal likelihood as

$$\mathcal{L}\left(q(\hat{\mathbf{S}}), \hat{\mathrm{T}}, \boldsymbol{\Theta}\right) = \int q(\hat{\mathbf{S}}) \log \frac{\tilde{p}(\mathbf{Y}|\hat{\mathbf{S}})p(\hat{\mathbf{S}})}{q(\hat{\mathbf{S}})} d\hat{\mathbf{S}}, \tag{5.18}$$

where

$$\begin{aligned}\log \tilde{p}(\mathbf{Y}|\hat{\mathbf{S}}) &= \int p(\mathbf{A}|\hat{\mathbf{A}})p(\mathbf{S}|\hat{\mathbf{S}})q(\hat{\mathbf{A}}) \log \frac{p(\mathbf{Y}|\mathbf{A},\mathbf{S})p(\hat{\mathbf{A}})}{q(\hat{\mathbf{A}})} d\mathbf{A} d\mathbf{S} d\hat{\mathbf{A}} \\ &= \langle \log p(\mathbf{Y}|\mathbf{A},\mathbf{S}) \rangle + \text{const} \\ &= \log \mathcal{N}\left(\mathbf{U}^{-1}\mathbf{z}_{:} \left| \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \hat{\mathbf{s}}_{:}, \mathbf{U}^{-1}\right.\right) - \frac{1}{2} \operatorname{tr}\left(\operatorname{cov}\left(\mathbf{s}_{:}|\hat{\mathbf{S}}\right)\mathbf{U}\right) + \text{const},\end{aligned}$$

and const is constant with respect to $\hat{\mathbf{S}}$, $\hat{\mathrm{T}}$ and $\boldsymbol{\Theta}$. The vector $\mathbf{z}_{:}$ and matrix $\mathbf{U}$ are defined in (5.10) and (5.11), respectively. The covariance term $\operatorname{cov}\left(\mathbf{s}_{:}|\hat{\mathbf{S}}\right)$ is defined similarly as in (4.20).

The optimal $q(\hat{\mathbf{S}})$ can be found by interpreting the lower bound (5.18) as the Kullback-Leibler divergence between $q(\hat{\mathbf{S}})$ and $\tilde{p}(\mathbf{Y}|\hat{\mathbf{S}})p(\hat{\mathbf{S}})$. Thus, the optimal $q(\hat{\mathbf{S}})$ is proportional to $\tilde{p}(\mathbf{Y}|\hat{\mathbf{S}})p(\hat{\mathbf{S}})$, resulting in

$$\begin{aligned} q(\hat{\mathbf{S}}) &= \mathcal{N}\left(\hat{\mathbf{s}}_{:} \left| \mathbf{V}\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\mathbf{z}_{:}, \mathbf{V}\right.\right), \text{ where} \\ \mathbf{V} &= \left(\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} + \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\mathbf{U}\mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}}\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1}\right)^{-1}, \end{aligned} \tag{5.19}$$

which can be interpreted as the posterior in a model with a likelihood $\tilde{p}(\mathbf{Y}|\hat{\mathbf{S}})$ and a prior $p(\hat{\mathbf{S}})$. Again, the relation to the previously derived posterior distributions is clearly visible. It is identical to the sparse posterior in regular GP in equation (4.21), except the observations are $\mathbf{U}^{-1}\mathbf{z}_{:}$ with a noise covariance $\mathbf{U}^{-1}$. Compared to the variational approximate posterior (5.12) in GPFA, the observations and noise covariance are projected to the lower dimensional space of the auxiliary variables $\hat{\mathbf{S}}$.

5.3.2 Component-wise factorization

In order to further reduce the computational cost, the components can be factorized similarly as in Section 5.2.2:

$$q(\hat{\mathbf{S}}) = \prod_{d=1}^{D} q(\hat{\mathbf{s}}_{d:}).$$

This results in the following approximate posterior distributions:

$$\begin{aligned} q(\hat{\mathbf{s}}_{d:}) &= \mathcal{N}\left(\hat{\mathbf{s}}_{d:} \left| \mathbf{V}_d\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\mathbf{z}_d, \mathbf{V}_d\right.\right), \text{ where} \\ \mathbf{V}_d &= \left(\mathbf{K}_{\hat{\mathbf{s}}_{d:}}^{-1} + \mathbf{K}_{\hat{\mathbf{s}}_{d:}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{d:},\mathbf{s}_{d:}}\mathbf{U}_d\mathbf{K}_{\mathbf{s}_{d:},\hat{\mathbf{s}}_{d:}}\mathbf{K}_{\hat{\mathbf{s}}_{d:}}^{-1}\right)^{-1}. \end{aligned} \tag{5.20}$$

The vector $\mathbf{z}_d$ and the diagonal matrix $\mathbf{U}_d$ are defined in (5.13) and (5.14), respectively. Note that the sparse approximation can be done independently to

each component. For instance, if some of the components have a long length scale but the others have very short length scale, one can apply the sparse approximation to the slow components and compactly supported covariance functions to the fast components, thus enabling efficient inference on both types of components.

5.3.3 Learning the hyperparameters

The maximum likelihood estimates for the hyperparameters $\boldsymbol{\theta}_d$ can be found by maximizing the lower bound of the log marginal likelihood. The lower bound is obtained by inserting the normalized $q(\hat{\mathbf{S}}) = \tilde{p}(\mathbf{Y}|\hat{\mathbf{S}})p(\hat{\mathbf{S}})/\int \tilde{p}(\mathbf{Y}|\hat{\mathbf{S}})p(\hat{\mathbf{S}})d\hat{\mathbf{S}}$ to the lower bound (5.18), yielding

$$\begin{aligned}\mathcal{L}\left(\hat{\mathrm{T}}, \boldsymbol{\Theta}\right) &= \log \int \tilde{p}(\mathbf{Y}|\hat{\mathbf{S}})p(\hat{\mathbf{S}})d\hat{\mathbf{S}}, \\ &= \log \mathcal{N}\left(\mathbf{U}^{-1}\mathbf{z}_{:} \left| \mathbf{0}, \mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}}\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\right.\right) \\ &\quad - \frac{1}{2}\operatorname{tr}\left(\operatorname{cov}\left(\mathbf{s}_{:}|\hat{\mathbf{S}}\right)\mathbf{U}\right) + \text{const.}\end{aligned} \tag{5.21}$$

The lower bound for the approximation using component-wise factorization is similarly

$$\begin{aligned}\mathcal{L}\left(\hat{\mathrm{T}}_d, \boldsymbol{\theta}_d\right) &= \log \mathcal{N}\left(\mathbf{U}_d^{-1}\mathbf{z}_d \left| \mathbf{0}, \mathbf{U}_d^{-1} + \mathbf{K}_{\mathbf{s}_{d:},\hat{\mathbf{s}}_{d:}}\mathbf{K}_{\hat{\mathbf{s}}_{d:}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{d:},\mathbf{s}_{d:}}\right.\right) \\ &\quad - \frac{1}{2}\operatorname{tr}\left(\operatorname{cov}\left(\mathbf{s}_{d:}|\hat{\mathbf{s}}_{d:}\right)\mathbf{U}_d\right) + \text{const.}\end{aligned} \tag{5.22}$$

These bounds can be used for optimizing the inducing inputs $\hat{\mathrm{T}}$ and the hyperparameters $\boldsymbol{\theta}_d$ of the covariance functions k_{s_d}. Evaluation of the lower bounds and their gradients are discussed in Appendix B.2.

5.4 Variational approximations for A and $\boldsymbol{\tau}$

The update rules for $q(\mathbf{A})$ and $q(\boldsymbol{\tau})$ are evaluated as follows. Due to symmetry in the model, the optimal $q(\mathbf{A})$ can be computed similarly as $q(\mathbf{S})$ in Sections 5.2 and 5.3 by exchanging $\mathbf{A}$ and $\mathbf{S}$ appropriately. The update rule of the factor $q(\boldsymbol{\tau})$ equals

$$\begin{aligned}q(\boldsymbol{\tau}) &= \prod_{mn\in\mathcal{O}} \mathcal{G}\left(\tau_{mn} \left| \bar{a}_{\tau_{mn}}, \bar{b}_{\tau_{mn}}\right.\right), \\ \bar{a}_{\tau_{mn}} &= a_\tau + \frac{1}{2}, \\ \bar{b}_{\tau_{mn}} &= b_\tau + \frac{1}{2}\left\langle \left(y_{mn} - \mathbf{a}_{m:}^{\mathrm{T}}\mathbf{s}_{:n}\right)^2 \right\rangle,\end{aligned}$$

where $\mathcal{O}$ is the set of indices (m, n) for which the corresponding observation y_{mn} is not missing. If a common noise level is used for several observations, the formulas are changed by adding summations over the corresponding indices. For instance, if isotropic noise is used ($\tau_{mn} = \tau$), the update rule equals

$$\begin{aligned} q(\tau) &= \mathcal{G}\left(\tau \left| \bar{a}_\tau, \bar{b}_\tau \right.\right), \\ \bar{a}_\tau &= a_\tau + \frac{1}{2} \sum_{mn \in \mathcal{O}} 1, \\ \bar{b}_\tau &= b_\tau + \frac{1}{2} \sum_{mn \in \mathcal{O}} \left\langle \left(y_{mn} - \mathbf{a}_{m:}^{\mathrm{T}} \mathbf{s}_{:n} \right)^2 \right\rangle . \end{aligned}$$

The evaluation of the parameter $\bar{b}_\tau$ costs $O(MND^2)$, which can be reduced to $O(MND)$ by using component-wise factorization. The variational learning algorithm consists of alternate updates of the factors $q(\mathbf{A})$, $q(\mathbf{S})$ and $q(\boldsymbol{\tau})$ until convergence.

5.5 Comments on implementation

The presented variational approximations differ in their computational cost. Using sparse approximations and component-wise factorizations helps in improving the speed of learning, as shown in Table 5.1. However, there is always a trade-off when improving the speed with further approximations and factorizations: the accuracy of the approximation decreases and the posterior dependencies between the variables are lost because they are restricted to be independent. If the posterior dependencies are important to the application, the component-wise factorization should be avoided unless the computational cost is too high without the factorization. On the other hand, one may be interested in finding decoupled components, and thus the component-wise factorization may actually help because the approximation favors independent components. The effect of these approximations are not further examined in this thesis, but for more detailed discussion on the effect of factorial approximations in the variational framework, refer to, for instance, the paper by Ilin and Valpola (2005).

Appendix B gives detailed derivations and formulas for evaluating the lower bounds, gradients and approximate posterior distributions efficiently. In the following paragraphs, we will discuss some of the main issues without going to the details of the evaluations.

The presented learning algorithms, and Gaussian processes in general, require inverting covariance matrices, which can be implemented efficiently and in a numerically stable way by utilizing the Cholesky decomposition. It decomposes a symmetric positive-definite matrix $\boldsymbol{\Sigma}$ as $\mathbf{L}\mathbf{L}^{\mathrm{T}} = \boldsymbol{\Sigma}$, where $\mathbf{L}$ is a

Method	Approximation		Complexity
GP to $\mathbf{Y}$			$O(N^3M^3)$
GPFA	$q(\mathbf{S})$	(5.12)	$O(D^3N^3 + D^3M^3)$
GPFA	$q(\mathbf{s}_{d:})$	(5.15)	$O(DN^3 + DM^3)$
GPFA	$q(\hat{\mathbf{S}})$	(5.19)	$O((\sum_{d=1}^{D} \hat{N}_d)^2 N + (\sum_{d=1}^{D} \hat{M}_d)^2 M)$
GPFA	$q(\hat{\mathbf{s}}_{d:})$	(5.20)	$O(\sum_{d=1}^{D} \hat{N}_d^2 N + \sum_{d=1}^{D} \hat{M}_d^2 M)$

Table 5.1: The computational complexity of different algorithms. In addition, the evaluation of the distribution $q(\boldsymbol{\tau})$ costs $O(MND^2)$, or $O(MND)$ if component-wise factorization is used.

lower-triangular matrix. The computational cost of finding the decomposition scales cubically with respect to the dimensionality of the matrix, that is, the number of columns and rows. For sparse covariance matrices produced by compactly supported covariance functions, the decomposition can be evaluated more efficiently because the sparsity is preserved in the decomposition. After decomposing a matrix $\boldsymbol{\Sigma}$, the lower triangular matrix $\mathbf{L}$ can be used for efficiently evaluating particular terms such as $\mathbf{z}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{y}$ and $\log|\boldsymbol{\Sigma}|$.

The different lower bounds of the log marginal likelihood and their gradients contain trace terms, which can be evaluated quite efficiently. The trace of the product of two matrices $\mathbf{A}$ and $\mathbf{B}$ can be evaluated as a dot product of their elements, that is, $\mathrm{tr}(\mathbf{AB}) = [\mathbf{A}]_{:}^{\mathrm{T}} \left[\mathbf{B}^{\mathrm{T}}\right]_{:}$. If some elements are zero in one of the matrices, there is no need to evaluate the corresponding elements of the other matrix. This property makes it feasible to optimize the hyperparameters in sparse approximations or compactly supported covariance functions.

In the case of the sparse approximation, the latter term in the lower bounds (5.21) and (5.22) simplifies as

$$\mathrm{tr}\left(\mathrm{cov}\left(\mathbf{s}_{:}|\hat{\mathbf{S}}\right)\mathbf{U}\right) = \mathrm{tr}\left(\sum_{n=1}^{N}\mathrm{cov}\left(\mathbf{s}_{:n}|\hat{\mathbf{S}}\right)\mathbf{U}_n\right) = \sum_{n=1}^{N}\sum_{d=1}^{D}\mathrm{var}(\mathbf{s}_{dn}|\hat{\mathbf{s}}_{d:})\left[\mathbf{U}_n\right]_{dd},$$

which follows from the sparse structure of $\mathbf{U}$ and a priori independence of components. This can also be seen directly by recalling that $\mathbf{K}_{\mathbf{s}_{:}}$ and $\mathbf{U}$ have overlapping nonzero elements only on the main diagonal.

When using compactly supported covariance functions, the term with the most concern is $\mathrm{tr}((\mathbf{K}_{\mathbf{s}_{:}} + \mathbf{U}^{-1})^{-1}\frac{\partial \mathbf{K}_{\mathbf{s}_{:}}}{\partial \theta_d})$ in the gradient, as shown in (B.1). The inverse of a sparse matrix is not sparse in general, thus evaluating the full inverse has a high computational cost. However, because $\frac{\partial \mathbf{K}_{\mathbf{s}_{:}}}{\partial \theta}$ is sparse by construction, only a small number of the elements from the inverse matrix $(\mathbf{K}_{\mathbf{s}_{:}} + \mathbf{U}^{-1})^{-1}$ are needed in order to evaluate the trace. For evaluating only some elements of the inverse of a sparse matrix, we utilized an efficient implementation by Vanhatalo and Vehtari (2008). The implementation is based

on an algorithm introduced by Takahashi et al. (1973) and discussed in more detail by Niessner and Reichert (1983).

Since the model is nonlinear, learning can be sensitive to initial conditions. If the hyperparameters are initialized poorly, or they are updated too early or too late, the algorithm might find inferior ML estimates potentially resulting in pruning out of relevant components or otherwise poor hyperparameter values for the GPs. For instance, a badly initialized period of a periodic component can be hard to infer from the data if several other components were also learned. Regularization of the hyperparameters could help in improving the learning, and it also seemed that the hyperparameters should not be updated before a few iterations of the learning algorithm is completed. However, this thesis did not study the effects of the initialization exhaustively. In order to reduce the effect of the initialization, it is possible to use several random initializations and choose the best solution based on the VB lower bound of the marginal likelihood.

5.6 Related work

Gaussian processes offer a flexible and sophisticated way of setting priors for the loadings or states in the FA model. Teh et al. (2005) and Yu et al. (2009) set Gaussian process priors over the states $\mathbf{S}$ and use a maximum likelihood estimate for the loadings $\mathbf{A}$. However, spatio-temporal models should not ignore the spatial structure. Typically, spatial structure has been modeled with Gaussian processes, also known as kriging in the context of spatial interpolation (Cressie, 1993). Lopes et al. (2008) use GPs as the spatial prior but resort to AR models as the temporal prior.

Although GPs have been applied to the states and the loadings separately in different models, using GP priors for both of the variables at the same time is quite a recent approach. Schmidt and Laurberg (2008) assigned Gaussian process priors for both variables in a bit more general context of nonnegative matrix factorization. However, they use MAP estimates for both of the variables, thus ignoring all the uncertainty.

Recently, Schmidt (2009) presented a model which is similar to the GPFA model presented in this thesis. Whereas GPFA divides the input space into two subspaces (spatial and temporal), their model divides the input space $\mathcal{X}$ into arbitrary number C of subspaces $\{\mathcal{X}_c \subset \mathcal{X}\}_{c=1}^{C}$. Similarly, they use D independent Gaussian processes over each of the subspaces:

$$y(\mathbf{x}) = \sum_{d=1}^{D} \left[s_{1d}(\mathbf{x}_1) s_{2d}(\mathbf{x}_2) \cdots s_{Cd}(\mathbf{x}_C) \right], \tag{5.23}$$

where s_{cd} is a function over the subspace $\mathcal{X}_c$, and the input $\mathbf{x}_c \in \mathcal{X}_c$ is the value of the original input $\mathbf{x} \in \mathcal{X}$ within the subspace $\mathcal{X}_c$. They do inference with MCMC, which does not usually scale well to large problems. Therefore, the variational framework has significant computational advantages as the sparse approximations and the component-wise factorizations can be done in a principled way in order to reduce the computational burden.

5.7 Conclusions

This chapter introduced a novel model for spatio-temporal modeling. The model can be seen as a mixture of factor analysis, temporal smoothing and kriging. Thus, the model is essentially a linear latent variable model with spatially and temporally structured priors. The model has a significant advantage over standard GP regression over the spatio-temporal domain because the computational cost is remarkably reduced by dividing the input space into spatial and temporal subspaces. The computational cost was further reduced by exploiting sparse approximations and component-wise factorizations. We also briefly discussed how to efficiently implement the evaluations in the learning algorithm. The discussion on related work showed that the model extends very recent previous works.

Chapter 6

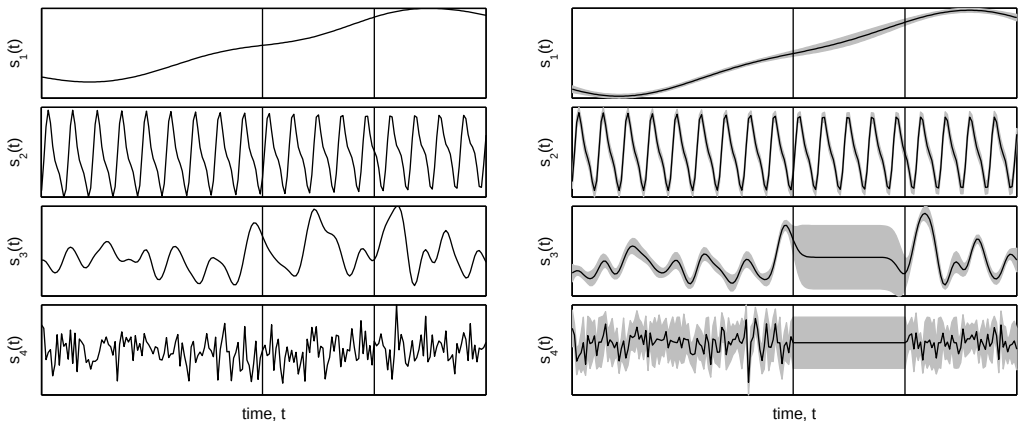
Experiments

This chapter presents experimental results with the GPFA model introduced in Chapter 5. Section 6.1 shows the results of learning the model based on a dataset generated artificially with the same model. The results are evaluated by comparing the extracted components to the real components and illustrating the accuracy of predictions. Section 6.2 presents a challenging real-world problem of modeling a noisy global sea-surface temperature dataset. The performance is compared to variational Bayesian PCA as a baseline.

6.1 Artificial example

An artificial dataset was generated by using the GPFA model. Four ($D = 4$) latent spatial and temporal components $\mathbf{a}_{:d}$ and $\mathbf{s}_{d:}$ were generated by taking samples from the GP priors. The priors of the four temporal components used different covariance functions in order to have features with different characteristics. A slowly changing component was generated by using the squared exponential covariance function (4.12). An approximately periodic signal used the quasi-periodic covariance function (4.15), which is the product of the squared exponential (4.12) and the periodic covariance function (4.13). A smoothly varying component was generated by using the squared exponential (4.12). A fast changing component was generated with the compactly supported covariance function (4.14) using a short length scale. The generated sources are shown in Figure 6.1a. The spatial loadings $\mathbf{a}_{:d}$ used the scaled squared exponential covariance function (4.16) with different length scales. These loadings are shown in Figure 6.2a.

The observations y_{mn} were generated by using a relatively small amount of isotropic noise. The standard deviation of the noise was 1.0, whereas the observation signals before adding the noise had standard deviation of 7.6 on average. The number of spatial locations and time instances was set to $M = 30$



(a) (b)

Figure 6.1: The latent time series $s_d(t)$ in the artificial experiment. (a) The true latent signals used to generate the data. (b) The posteriors of the four latent signals. The solid lines show the posterior mean and gray color shows two standard deviations. The gap with no training observations is marked with vertical lines.

and $N = 200$, respectively. The spatial locations were chosen randomly from a uniform distribution over a two-dimensional rectangular area. The time instances were deterministic and uniformly spaced.

Most of the data points y_{mn} were marked as missing. Observations from all locations were removed for a significant time period. This resulted in a gap in the data. In addition, 90% of the remaining observations were randomly removed. Therefore, only 452 noisy observations remained in total.

We trained the GPFA model on the generated noisy data. We used the same covariance functions that were used to generate the data. The hyperparameters of the covariance functions were initialized randomly close to the values used for data generation, assuming that a good guess about the hidden signals can be obtained by exploratory analysis of data. For the approximate posterior distribution, no component-wise factorization nor sparse approximations were used (see Section 5.2.1 for details).

Figure 6.1 shows the latent sources $s_d(t)$ which were used to generate the data and recovered by the proposed algorithm. Note that the algorithm separated four latent signals with the different variability time scales. For the slow and periodic components, the observation gap is recovered with high precision, whereas the two fast components have high uncertainty over the gap. The fourth component has high uncertainty also outside the gap.

Figure 6.2 shows the spatial loading functions $a_d(l)$ used to generate the data and recovered by the proposed algorithm. The loadings are recovered

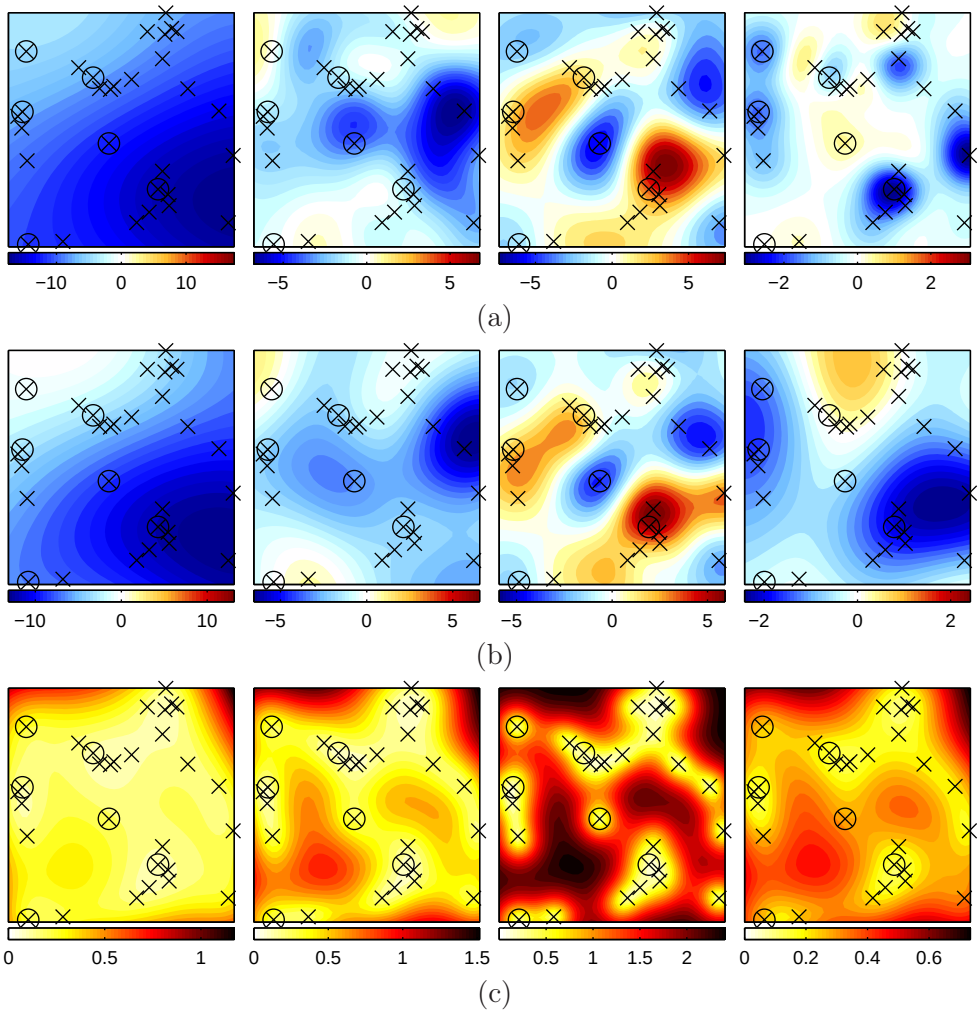


Figure 6.2: The spatial loading functions $a_d(l)$ in the artificial experiment: (a) the true loadings used to generate the data; (b) the posterior means; (c) the standard deviations computed from the posterior. The crosses show the locations of the observations, and the circled ones are examples for predictive evaluation in Figure 6.3.

accurately for the first three components, but the learned fourth component differs from the original component in smoothness: the length scale is estimated to be too large. This shows the disadvantage of using ML estimate because the estimate does not identify the uncertainty, which can be quite significant for the length scale of the fourth component.

Figure 6.3 shows the posterior predictive distributions for six sensors, located as shown in Figure 6.2. The test values are noisy data points y_{mn} which were removed before training. Although some of the sensors contain so few

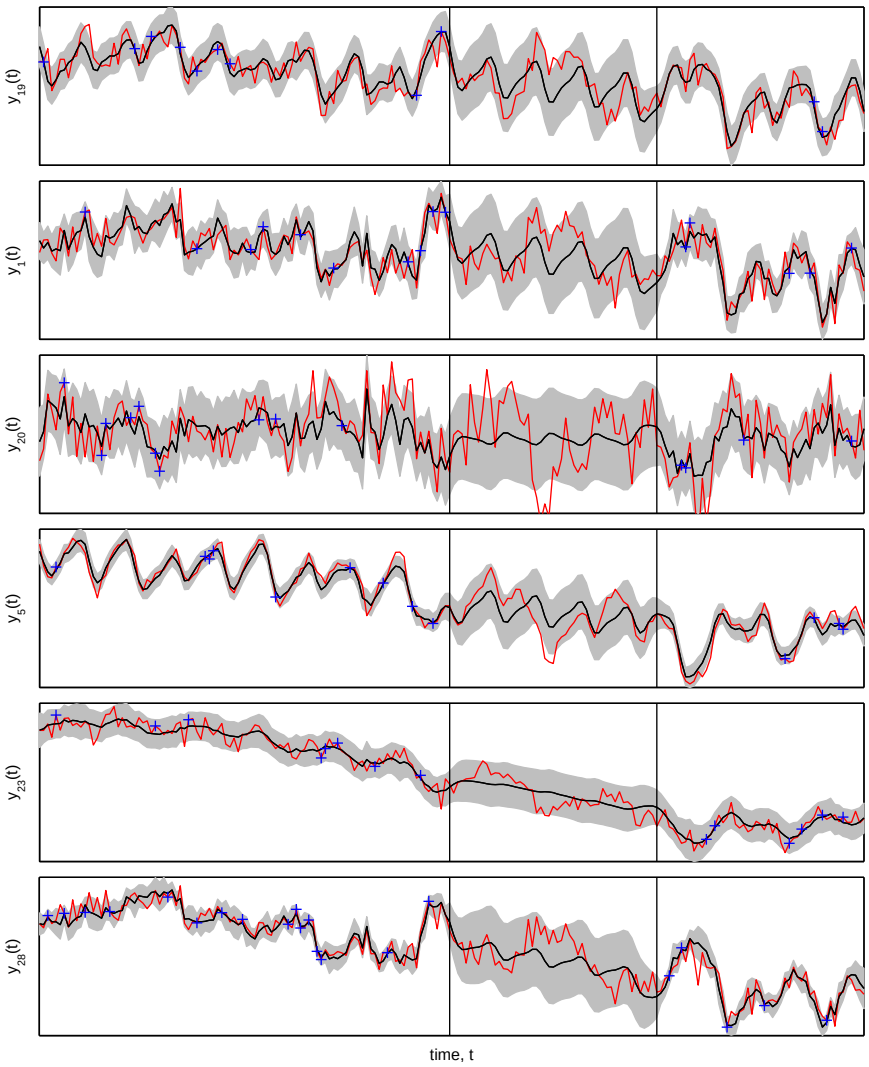
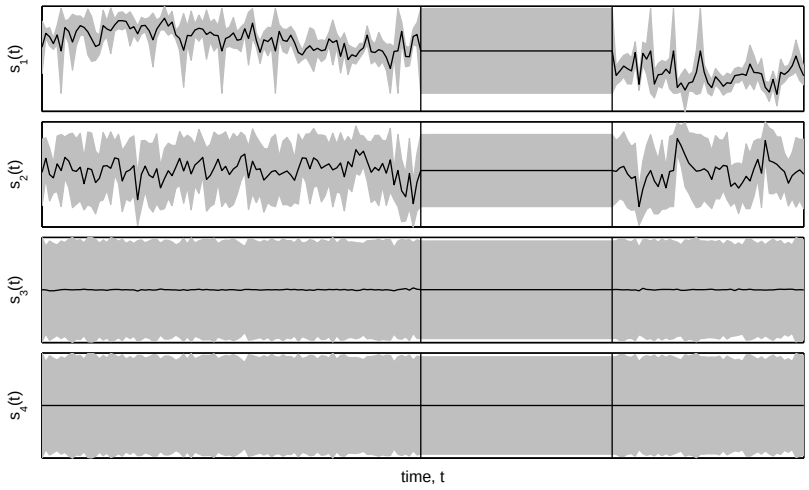


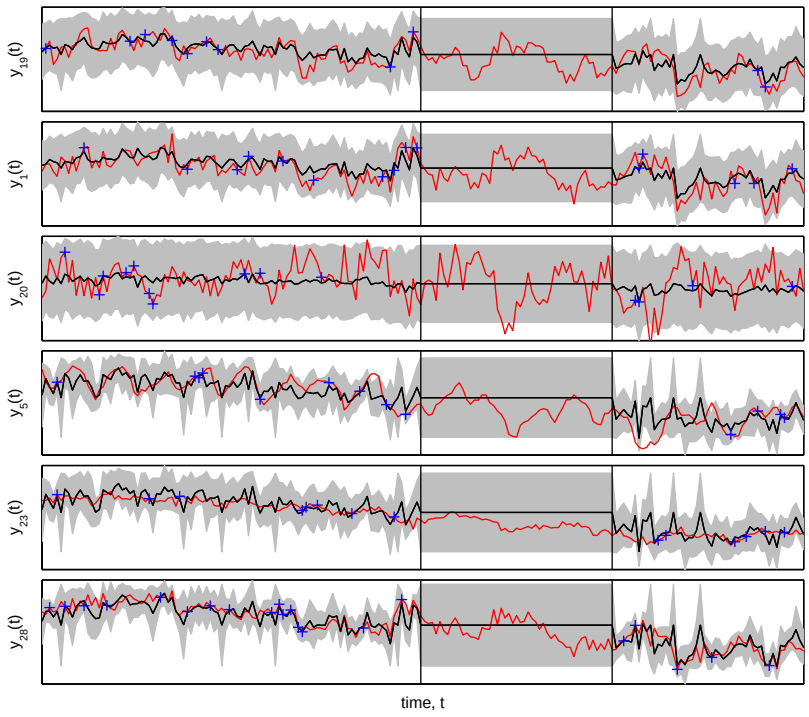
Figure 6.3: Posterior predictive distribution for six randomly selected locations. The training observations are shown as blue crosses and the noisy test data as solid red lines. The solid black lines and gray coloring show the mean and two standard deviations computed from the approximate posterior distribution. The gap with no training observations is marked with vertical lines.

observations that there is no evidence for complex temporal structure in those observations alone, the model has captured the complex structure and reconstructed the test values fairly well. This is a positive effect of the spatially smooth priors. In addition to the found structure, the uncertainty explains the errors in the predictions, which is an important property from the Bayesian viewpoint.

For comparison, Figure 6.4 shows the temporal components and the pre-



(a)



(b)

Figure 6.4: Results for the artificial dataset with VB-PCA. (a) The posteriors of the four latent signals. (b) Posterior predictive distribution for six randomly selected locations. The training observations are shown as blue crosses and the noisy test data as solid red lines. The solid black lines and gray coloring show the mean and two standard deviations computed from the approximate posterior distribution. The gap with no training observations is marked with vertical lines.

dictive distributions obtained by using variational Bayesian PCA (VB-PCA) (Bishop, 1999b). Clearly, VB-PCA is unable to learn the complex structure from the sparse data: two of the latent temporal components are pruned out, and the predictive distributions are extremely noisy.

6.2 Reconstruction of global sea surface temperature

This section demonstrates how the presented model can be applied to analyze and reconstruct global sea surface temperatures (SST) by using U.K. Meteorological Office historical SST dataset (MOHSST5) (Bottomley et al., 1990). The dataset contains monthly SST anomalies in the 1856-1991 period for $5^\circ \times 5^\circ$ longitude-latitude bins. Thus, the number of time instances and spatial locations is approximately 1600 and 1700, respectively. The dataset is sparse, especially during the 19th century and the World Wars, and near the polar regions. The dataset consists of more than 10^6 observations in total, thus having 55% of the values missing.

GPFA was used to estimate $D = 80$ components, which is the same number as used by Kaplan et al. (1998) for the same problem. We withdrew 20% of the data from the training set and used this part for testing the reconstruction accuracy. Five time signals $\mathbf{s}_{d:}$ used the squared exponential covariance function (4.12) to describe climate trends. Another five components also used the squared exponential function to model prominent interannual phenomena such as El Niño. Five temporal components were modeled with the quasi-periodic covariance function (4.15) to capture periodic signals (e.g., related to the annual cycle). Finally, the piecewise polynomial functions (4.14) were used to describe the remaining 65 time signals. These dimensionalities were chosen ad hoc. The spatial patterns $\mathbf{a}_{:d}$ were modeled with the scaled squared exponential (4.16). The additional scale parameter θ_2 in (4.16) helps in pruning out unnecessary components. The distance r between the locations l_i and l_j was measured on the surface of the Earth using the spherical law of cosines (see Appendix A.2 for details).

The posterior was approximated by using the component-wise factorization. 500 inducing inputs were also introduced for each spatial function $a_d(l)$ in order to use sparse variational approximations. Similar sparse approximations were used for the 15 temporal functions $s_d(t)$ which modeled slow climate variability: the slowest, interannual and quasi-periodic components had 80, 300 and 300 inducing inputs, respectively. The inducing inputs were initialized by taking a random subset from the original inputs and then kept fixed throughout learning because their optimization would have increased the computational burden substantially. The rest of the temporal components

allowed efficient computations by using the piecewise polynomial covariance function (4.14), which produces sparse covariance matrices.

The dataset was preprocessed by weighting the data points by the square root of the corresponding latitudes in order to diminish the effect of denser sampling in the polar regions. The weight for the m-th row of $\mathbf{Y}$ was set to

$$\omega_m = \sqrt{\cos(\phi_m)},$$

where ϕ_m is the latitude of the corresponding location. The weighting can also be seen as using a spatially varying noise level (i.e., $\tau_m = \omega_m^2 \tau$).

The GP hyperparameters were initialized by taking into account the assumed smoothness of the spatial patterns and the variability timescale of the temporal factors. The length scales of the slow, interannual and fast components were initialized randomly to 10–20 years, 2–10 years, and 0.5–1 year. The periods of the periodic components were initialized randomly between 0.5 and 1.5 years. The factors $\mathbf{S}$ were initialized randomly by sampling from the prior and the weights $\mathbf{A}$ were initialized to zero. The variational Bayesian EM-algorithm of GPFA was run for 200 iterations. For comparison, VB-PCA was applied to the same dataset.

The principal components can be found by rotating the latent subspace such that the variables are orthogonal and account for the most variance in descending order. This is achieved in two steps: (1) the latent states $\mathbf{S}$ are centered by subtracting the expectation of the row-wise mean, and (2) $\mathbf{S}$ and $\mathbf{A}$ are rotated such that $\frac{1}{N}\left\langle \mathbf{S}\mathbf{S}^{\mathrm{T}} \right\rangle = \mathbf{I}$ and $\left\langle \mathbf{A}^{\mathrm{T}}\mathbf{A} \right\rangle$ is diagonal.

Figure 6.5 shows the spatial and temporal patterns of the four most dominant principal components for both models. Note that the principal components of GPFA are mixtures of the latent GP components. The GPFA principal components and the corresponding spatial patterns are generally smoother, especially in the data-sparse regions, for example, in the period before 1875. The spatial VB-PCA components are noisy near the South Pole, whereas the GPFA components are much smoother. The first and the second principal components of GPFA as well as the first and the third components of VB-PCA are related to El Niño. The fourth temporal component in VB-PCA has a period corresponding to the yearly oscillation but none of the four GPFA components contains such a strong yearly oscillation. Note that the uncertainty in the temporal VB-PCA components is much higher in the period before 1875. Thus, the GPFA model has learned some structure which helps in reconstructing the sparse parts of the data.

The importance of the latent GP components can be examined by comparing the amount of data variance explained by each component, as shown in Figure 6.6. We estimate the amount of variance each component explains

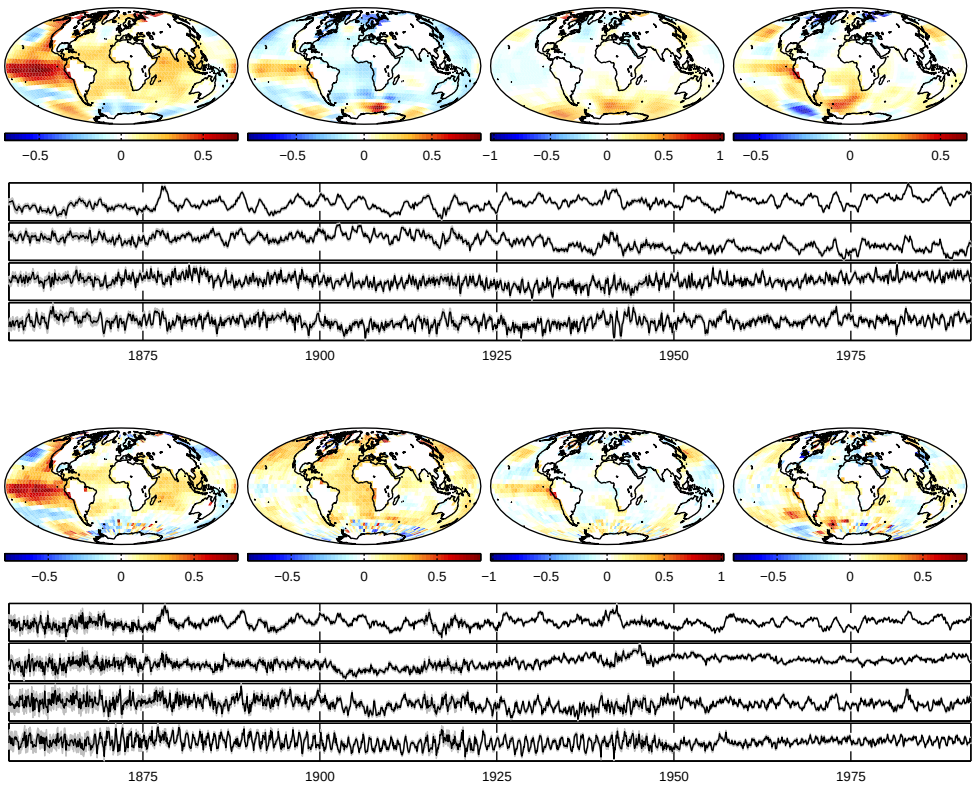


Figure 6.5: Experimental results for the MOHSST5 dataset. The spatial and temporal patterns of the four most dominating principal components for GPFA (above) and VB-PCA (below). The solid lines and gray color in the time series show the mean and two standard deviations of the posterior distribution. The uncertainty of the spatial patterns are not shown, and we saturated the visualizations of the VB-PCA components to reduce the effect of the uncertain pole regions.

by the corresponding diagonal element of

$$\frac{1}{MN} \left\langle \mathbf{S}\mathbf{S}^{\mathrm{T}} \right\rangle \left\langle \mathbf{A}^{\mathrm{T}}\mathbf{A} \right\rangle ,$$

where the row-wise mean has been subtracted from $\mathbf{S}$. The GPFA model efficiently used only some of the 15 slow components: about three slow and only one interannual components explained relatively large amounts of the variance. Thus, unnecessary components were pruned out. Interestingly, the periodic components have very small contribution to the data variance. This may relate to the fact that the yearly oscillation had been eliminated from the dataset. Figure 6.7 shows the four GPFA components explaining the most variance. The most dominating component is clearly a smoothed version of

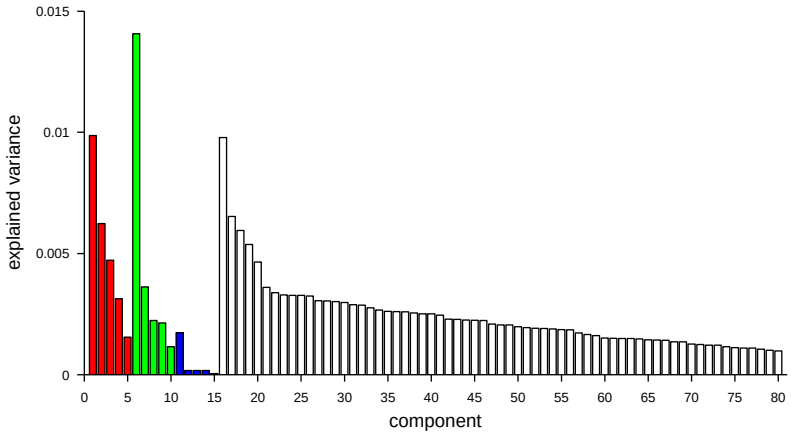


Figure 6.6: The amount of data variance explained by each GPFA component. The components 1–5 are slow (red), 6–10 interannual (green), 11–15 periodic (blue) and 16–80 fast (white). The components are ordered within each group with respect to the explained variance.

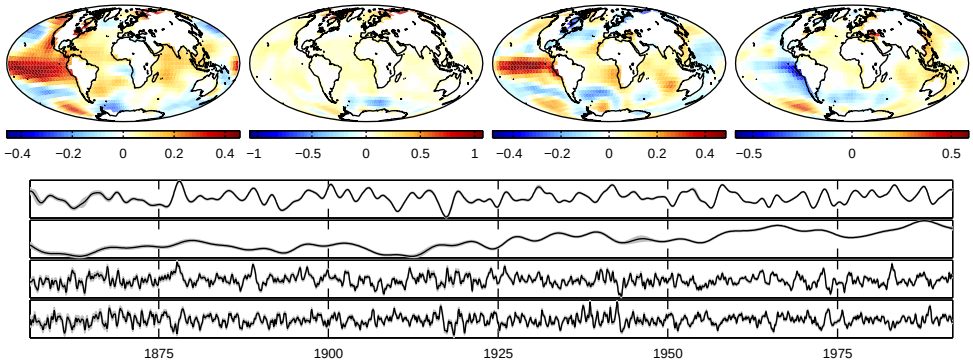


Figure 6.7: The four GPFA components explaining the most data variance.

the El Niño component. The second component is a smooth trend. The third component and the negative of the fourth component are also similar to the E Niño component.

Finally, we compared the two models by computing the weighted root mean square reconstruction error on the test set:

$$E_{\mathrm{RMSE}} = \sqrt{\sum_{mn \in \tilde{\mathcal{O}}} \omega_m^2 \left(\tilde{y}_{mn} - \langle \mathbf{a}_{m:} \rangle^{\mathrm{T}} \langle \mathbf{s}_{:n} \rangle \right)^2 \Big/ \sum_{mn \in \tilde{\mathcal{O}}} \omega_m^2},$$

where $\tilde{y}_{mn}$ are the observations in the test set, and the set $\tilde{\mathcal{O}}$ contains the indices of those observations. The prediction errors were 0.5714 for GPFA and

0.6180 for VB-PCA. The improvement obtained by GPFA can be considered quite significant taking into account the substantial amount of noise in the data. However, the results might be improved by selecting more realistic covariance functions or using problem-specific prior knowledge to regularize the hyperparameters.

6.3 Conclusions

This chapter presented two experiments with the GPFA model. The artificial experiment illustrated how the model can find latent spatial and temporal structure. In the real-world experiment, GPFA was compared to VB-PCA in the reconstruction of sea surface temperatures. The results showed that the spatio-temporal structure in the priors can improve the modeling and extraction of latent sources.

Chapter 7

Conclusions

This thesis presented a novel spatio-temporal factor analysis model using Gaussian process priors over the spatial and temporal components. The model enables modeling and exploratory analysis of large spatio-temporal systems by utilizing efficient variational Bayesian approximations. The model was successfully applied to artificial and real-world datasets in the experimental section showing some benefits of using spatially and temporally structured priors. This chapter discusses some issues related to the model and future directions.

Although the sparse approximations make inference with Gaussian processes more feasible, they do not solve the computational issues completely. If the size of the dataset increases, it is likely that the number of inducing inputs must be increased with the same rate, that is, keeping the density of the inducing inputs fixed. Thus, the effective computational cost remains cubic with respect to the number of data points, because the sparse approximation only multiplies the computational cost with some small constant but does not change the asymptotic behavior. This makes the sparse approximations an infeasible solution for very large-scale problems unless extremely sparse approximations are used.

The presented model has an interesting special case, as it can be used for modeling local and global phenomena in univariate GP regression. An unknown function $y(t)$ can be decomposed as

$$y(t) = s_1(t) + s_2(t) + \text{noise},$$

where s_1 and s_2 correspond to slowly and fast varying components of the function y, respectively. Using the VB component-wise factorization, the components can be learned efficiently by using sparse approximations for s_1 and compactly supported covariance function for s_2. Vanhatalo and Vehtari (2008) presented similar idea using MCMC for inference. However, MCMC methods can be slow, and their model used approximate prior and likelihood function which are not guaranteed to bound the true marginal likelihood in any way.

The variational approach could potentially solve both of these problems, and the approximation might be extremely accurate because the assumption of independence between the slow and fast components sounds reasonable.

The model could possibly be improved by using separate modeling of global and local features. In order to capture short-scale local phenomena accurately, the GPFA model requires a large number of latent components. Instead of using a large number of components, it might be more reasonable to model these phenomena with some localized short-scale spatio-temporal Gaussian process. The local GP could use a covariance function that exploits the coupling of the spatial and temporal domains. Mathematically, such a model could be defined as

$$y(l,t) = r(l,t) + \sum_{d=1}^{D} a_d(l)s_d(t) + \text{noise},$$

where the new function $r(l,t)$ is a very short-scale spatio-temporal function. The global features could probably be modeled reasonably well with a small number D of components and by using very sparse approximations. A relevant question is how to model the short-scale function $r(l,t)$ efficiently.

In addition to compactly supported covariance functions, there exist other approaches for modeling short-scale phenomena in large datasets. For instance, Rasmussen and Ghahramani (2002) and Yuan and Neubauer (2009) have introduced mixtures of Gaussian process experts, and Snelson (2007) has suggested dividing the input space into clusters and modeling independent Gaussian processes over these small clusters. Although both the mixture modeling and the clustering approach may outperform compactly supported covariance functions and sparse approximations for some type of functions, it is not straightforward to apply them to the presented model within the variational Bayesian framework. Thus, the feasibility and benefits of those approaches remain an interesting open question.

The spatio-temporal climate modeling might be improved by taking into account the prior physical knowledge. Although the dynamics are usually very complex, utilizing even some simplified equations could make a remarkable difference. On the other hand, physical modeling could exploit the Bayesian modeling to handle uncertainty in order to learn properties which are not described well by physical models. Thus, combining physical and statistical modeling could benefit both frameworks opening a fascinating line of further research.

The model could be extended to have a robust noise model. This can be achieved by using Student-t as the noise distribution for each data dimension. The Student-t noise distribution does not significantly complicate the modeling, because the distribution can be seen as a Gaussian distribution using

spatio-temporally varying τ_{mn} with a Gamma prior. We have successfully applied such a robust noise model to VBPCA in a real-world application with badly corrupted temperature measurements (Luttinen et al., 2009a). Combining the spatio-temporal GPFA model with the robust noise model could improve the modeling of very noisy spatio-temporal datasets.

The noise model could also be extended hierarchically. The hierarchical noise model would allow the locations to have different noise levels while still being able to predict the noise level at new locations. In addition, temporally varying noise level might be reasonable when the dataset covers a very long time period, as measurements made one hundred years ago can be considered less accurate as the measurements made today using high-quality technology.

In the experimental section, we applied the presented model to the reconstruction of a historical sea surface temperature dataset. The current state-of-the-art reconstruction methods (Kaplan et al., 1998) are based on empirical orthogonal function (EOF) analysis and temporal smoothing. They are close to applying probabilistic PCA (Tipping and Bishop, 1999) to the data and fitting an autoregressive model to the posterior means of the latent temporal components. The presented GPFA model is based on similar modeling assumptions. The advantages of GPFA include: (1) uncertainty is handled properly by using the Bayesian framework; (2) the dimensionality reduction and smoothing are combined into one estimation procedure; and (3) by choosing the covariance functions properly, GP components can model spatial and temporal phenomena on different scales.

Future research includes performing more experiments with real-world data. These experiments could try using different covariance functions and regularizing the hyperparameters. In addition, GPFA should also be compared to other similar models in more detail. For instance, the state-space models described in Section 3.3 are relevant for such comparisons.

To summarize, the presented model gave promising results in the experiments and offers interesting directions for further research. We are planning to experiment with the model more exhaustively and study different ways to extend the model in order to improve its performance.

Bibliography

Attias, H., (2000). A variational Bayesian framework for graphical models. In: Solla, S., Leen, T., and Müller, K.-R. (eds.), *Advances in Neural Information Processing Systems 12*, pp. 209–215. MIT Press, Cambridge, MA, USA.

Banerjee, S., Carlin, B. P., and Gelfand, A. E., (2004). *Hierarchical Modeling and Analysis for Spatial Data*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Beal, M. J. and Ghahramani, Z., (2003). The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian Statistics 7* pp. 453–464.

Bishop, C., (1999a). Latent variable models. In: Jordan, M. (ed.), *Learning in Graphical Models*, pp. 371–403. The MIT Press, Cambridge, MA, USA.

Bishop, C. M., (1999b). Variational principal components. In: *Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN'99)*, pp. 509–514.

Bishop, C. M., (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2nd edn.

Bottomley, M., Folland, C. K., Hsiung, J., Newell, R. E., and Parker, D. E., (1990). *Global Ocean Surface Temperature Atlas*. Her Majesty's Stn. Off., Norwich, England.

Calder, C. A., (2007). Dynamic factor process convolution models for multivariate space-time data with application to air quality assessment. *Environmental and Ecological Statistics* 14, no. 3, 229–247.

Cressie, N., (1993). *Statistics for Spatial Data*. Wiley Series in Probability and Mathematical Statistics. Wiley, New York.

Cressie, N. and Huang, H.-C., (1999). Classes of nonseparable, spatio-temporal stationary covariance functions. *Journal of the American Statistical Association* 94, no. 448, 1330–1340.

Dempster, A. P., Laird, N. M., and Rubin, D. B., (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 39, no. 1, 1–38.

Finkenstädt, B., Held, L., and Isham, V. (eds.), (2007). *Statistical Methods for Spatio-Temporal Systems*. No. 107 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Gellert, W., Kustner, H., Hellwich, M., Kastner, H., Hirsch, K. A., and Reichardt, H., (1977). *The VNR Concise Encyclopedia of Mathematics*. Van Nostrand Reinhold.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B., (2003). *Bayesian Data Analysis*. Texts in Statistical Science. Chapman & Hall/CRC, Florida, 2nd edn.

Ghahramani, Z. and Roweis, S., (1999). Learning nonlinear dynamical systems using an EM algorithm. In: Kearns, M., Solla, S., and Cohn, D. (eds.), *Advances in Neural Information Processing Systems 11*, pp. 431–437. The MIT Press, Cambridge, MA, USA.

Grewal, M. S. and Andrews, A. P., (1993). *Kalman filtering: Theory and Practice*. Information and system science series. Prentice-Hall.

Honkela, A., Tornio, M., Raiko, T., and Karhunen, J., (2008). Natural conjugate gradient in variational inference. In: *Proceedings of the 14th International Conference on Neural Information Processing (ICONIP 2007)*, vol. 4985 of *Lecture Notes in Computer Science*, pp. 305–314, Kitakyushu, Japan. Springer-Verlag, Berlin.

Hyvärinen, A., Karhunen, J., and Oja, E., (2001). *Independent Component Analysis*. J. Wiley.

Ilin, A. and Raiko, T., (2008). Practical approaches to principal component analysis in the presence of missing values. Tech. Rep. TKK-ICS-R6, Helsinki University of Technology, Espoo, Finland. Available at http://www.cis.hut.fi/alexilin/.

Ilin, A. and Valpola, H., (2005). On the effect of the form of the posterior approximation in variational learning of ICA models. *Neural Processing Letters* 22, no. 2, 183–204.

Ilin, A., Valpola, H., and Oja, E., (2006). Exploratory analysis of climate data using source separation methods. *Neural Networks* 19, no. 2, 155–167.

Jaynes, E. T., (2003). *Probability Theory: The Logic of Science*. Cambridge University Press.

Jolliffe, I. T., (2002). *Principal Component Analysis*. Springer, 2nd edn.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K., (1998). An introduction to variational methods for graphical methods. In: *Machine Learning*, pp. 183–233. MIT Press.

Kaplan, A., Cane, M. A., Kushnir, Y., Clement, A. C., Blumenthal, M. B., and Rajagopalan, B., (1998). Analyses of global sea surface temperature 1856–1991. *Journal of Geophysical Research* 103, no. C9, 18567–18589.

Lopes, H. F., Salazar, E., and Gamerman, D., (2008). Spatial dynamic factor analysis. *Bayesian Analysis* 3, no. 4, 759–792.

Luttinen, J. and Ilin, A., (2009). Factor analysis with Gaussian process priors for modeling spatio-temporal data. In: *Advances in Neural Information Processing Systems 22*, Cambridge, MA. MIT Press. To appear.

Luttinen, J., Ilin, A., and Karhunen, J., (2009a). Bayesian robust PCA for incomplete data. In: *Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation (ICA'2009)*, pp. 66–73. Springer-Verlag.

Luttinen, J., Ilin, A., and Raiko, T., (2009b). Transformations for variational factor analysis to speed up learning. In: Verleysen, M. (ed.), *Proceedings of the 17th European Symposium on Artificial Neural Networks (ESANN'2009)*, pp. 77–82. d-side.

MacKay, D. J. C., (1998). Introduction to Gaussian processes. In: Bishop, C. M. (ed.), *Neural Networks and Machine Learning*, pp. 133–166. Springer.

MacKay, D. J. C., (1999). Comparison of approximate methods for handling hyperparameters. *Neural Computation* 11, 1035–1068.

Minka, T. P., (2001). Expectation propagation for approximate Bayesian inference. In: *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pp. 362–369. Morgan Kaufmann Publishers Inc.

Niessner, H. and Reichert, K., (1983). On computing the inverse of a sparse matrix. *International Journal for Numerical Methods in Engineering* 19, 1513–1526.

Orbanz, P., (2009). Construction of nonparametric Bayesian models from parametric Bayes equations. In: *Advances in Neural Information Processing Systems 22*, Cambridge, MA. MIT Press. To appear.

Pang, J., Qing, L., Huang, Q., Jiang, S., and Gao, W., (2007). Monocular tracking 3D people by Gaussian process spatio-temporal variable model. In: *Proceedings of the International Conference on Image Processing (ICIP'2007)*, pp. 41–44. IEEE.

Park, S. and Choi, S., (2007). Source separation with Gaussian process models. In: Kok, J. N., Koronacki, J., de Mántaras, R. L., Matwin, S., Mladenic, D., and Skowron, A. (eds.), *ECML*, vol. 4701 of *Lecture Notes in Computer Science*, pp. 262–273. Springer.

Petersen, K. B. and Pedersen, M. S., (2008). The matrix cookbook. Version 20081110.
URL: *http://www2.imm.dtu.dk/pubdb/p.php?3274*

Quiñonero-Candela, J. and Rasmussen, C. E., (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research* 6, 1939–1959.

Rasmussen, C. E. and Ghahramani, Z., (2002). Infinite mixtures of Gaussian process experts. In: Dietterich, T. G., Becker, S., and Ghahramani, Z. (eds.), *Advances in Neural Information Processing Systems 14*, pp. 881–888, Cambridge, MA. MIT Press.

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

Roweis, S. and Ghahramani, Z., (2001). An EM algorithm for identification of nonlinear dynamical systems. In: Haykin, S. (ed.), *Kalman Filtering and Neural Networks*, pp. 175–220. Wiley, New York.

Särelä, J. and Valpola, H., (2005). Denoising source separation. *Journal of Machine Learning Research* 6, 233–272.

Schmidt, M. N., (2009). Function factorization using warped Gaussian processes. In: Bottou, L. and Littman, M. (eds.), *Proceedings of the 26th International Conference on Machine Learning (ICML'09)*, pp. 921–928, Montreal. Omnipress.

Schmidt, M. N. and Laurberg, H., (2008). Nonnegative matrix factorization with Gaussian process priors. *Computational Intelligence and Neuroscience* 2008, 1–10.

Seeger, M., Williams, C. K. I., and Lawrence, N. D., (2003). Fast forward selection to speed up sparse gaussian process regression. In: Bishop, C. M. and Frey, B. J. (eds.), *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics (AISTATS'03)*, pp. 205–213. Society for Artificial Intelligence and Statistics.

Snelson, E., (2007). *Flexible and efficient Gaussian process models for machine learning*. Ph.D. thesis, Gatsby Computational Neuroscience Unit, University College London.

Snelson, E. and Ghahramani, Z., (2006). Sparse gaussian processes using pseudo-inputs. In: Weiss, Y., Schölkopf, B., and Platt, J. (eds.), *Advances in Neural Information Processing Systems 18*, pp. 1257–1264. MIT Press, Cambridge, MA.

Takahashi, K., Fagan, J., and Chen, M.-S., (1973). Formation of a sparse bus impedance matrix and its application to short circuit study. In: *Power Industry Computer Application Conference Proceedings*.

Teh, Y. W., Seeger, M., and Jordan, M. I., (2005). Semiparametric latent factor models. In: Cowell, R. G. and Ghahramani, Z. (eds.), *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS'05)*, pp. 333–340. Society for Artificial Intelligence and Statistics.

Tipping, M. E. and Bishop, C. M., (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B* 61, no. 3, 611–622.

Titsias, M. K., (2009). Variational learning of inducing variables in sparse Gaussian processes. In: van Dyk, D. and Welling, M. (eds.), *Proceedings of the 12th International Workshop on Artificial Intelligence and Statistics (AISTATS'09)*, pp. 567–574. Society for Artificial Intelligence and Statistics.

Tong, L., Soo, V., Liu, R., and Huang, Y., (1991). Indeterminacy and identifiability of blind identification. *IEEE Transactions on Circuits and Systems* 38, no. 5, 499–509.

Tong, Y. L., (1990). *The Multivariate Normal Distribution*. Springer.

Valpola, H. and Karhunen, J., (2002). An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural Computation* 14, no. 11, 2647–2692.

Vanhatalo, J. and Vehtari, A., (2008). Modelling local and global phenomena with sparse Gaussian processes. In: McAllester, D. A. and Myllymäki, P. (eds.), *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence (UAI'2008)*, pp. 571–578, Helsinki, Finland. AUAI Press.

von Storch, H. and Zwiers, W., (1999). *Statistical analysis in climate research*. Cambridge University Press, Cambridge, UK.

Williams, C. K. I. and Rasmussen, C. E., (1996). Gaussian processes for regression. In: Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E. (eds.), *Advances in Neural Information Processing Systems 8*, Cambridge, MA. MIT Press.

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., and Sahani, M., (2009). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In: Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems 21*, pp. 1881–1888. MIT Press, Cambridge, MA.

Yuan, C. and Neubauer, C., (2009). Variational mixture of Gaussian process experts. In: Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems 21*, pp. 1897–1904, Cambridge, MA. MIT Press.

Ziehe, A. and Müller, K.-R., (1998). TDSEP — an effective algorithm for blind separation using time structure. In: *Proceedings of the 8th International Conference on Artificial Neural Networks (ICANN '98)*, pp. 675–680, Skövde, Sweden.

Appendix A

Mathematical formulas

A.1 Matrix algebra

The matrix inversion lemmas are defined as

$$\left(\mathbf{A}+\mathbf{B}\mathbf{D}^{-1}\mathbf{C}\right)^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{D}+\mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\mathbf{C}\mathbf{A}^{-1}, \tag{A.1}$$

$$\left(\mathbf{A}+\mathbf{B}\mathbf{D}^{-1}\mathbf{C}\right)^{-1}\mathbf{B}\mathbf{D}^{-1}=\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{D}+\mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}, \tag{A.2}$$

where $\mathbf{A}$ and $\mathbf{D}$ are invertible square matrices, and $\mathbf{B}$ and $\mathbf{C}$ are matrices of appropriate size (see, e.g., Petersen and Pedersen, 2008). The matrix determinant lemma is defined as

$$\left|\mathbf{A}+\mathbf{B}\mathbf{C}\right|=\left|\mathbf{I}+\mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right|\cdot\left|\mathbf{A}\right|. \tag{A.3}$$

Derivatives for an invertible matrix $\mathbf{A}$ are defined as

$$d\log\left|\mathbf{A}\right|=\mathrm{tr}\left(\mathbf{A}^{-1}d\mathbf{A}\right), \tag{A.4}$$

$$d\mathbf{A}^{-1}=-\mathbf{A}^{-1}(d\mathbf{A})\mathbf{A}^{-1}. \tag{A.5}$$

For more details on the derivatives of matrices, refer to the collection of matrix-related formulas by Petersen and Pedersen (2008).

A.2 Distance measure on the Earth

Let two locations on Earth be denoted as $l_1=(\phi_1,\lambda_1)$ and $l_2=(\phi_2,\lambda_2)$, where ϕ_i and λ_i are the latitudes and longitudes, respectively. The distance between these locations can be evaluated by using the spherical law of cosines (Gellert et al., 1977):

$$d(l_1,l_2)=R\arccos(\sin\phi_1\sin\phi_2+\cos\phi_1\cos\phi_2\cos(\lambda_1-\lambda_2),$$

where $R\approx 6370\text{km}$ is the radius of the Earth. Note that the formula approximates the Earth as a sphere instead of an ellipsoid.

A.3 Conditional Gaussian distribution

Let two multivariate Gaussian variables $\mathbf{y}_1$ and $\mathbf{y}_2$ be distributed jointly as

$$\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}\right),$$

where $\boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{21}^{\mathrm{T}}$ because of the symmetricity of covariance matrices. The conditional distribution $p(\mathbf{y}_1|\mathbf{y}_2)$ then equals (see, e.g., Tong, 1990)

$$p(\mathbf{y}_1|\mathbf{y}_2) = \mathcal{N}\left(\mathbf{y}_1 \,\middle|\, \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\right). \tag{A.6}$$

Appendix B

Implementation of the model

B.1 Full approximation

This section explains how to implement the evaluation of the lower bounds, their gradients, and the approximate posterior distribution $q(\mathbf{S})$ for the variational approximation presented in Section 5.2.

The lower bounds (5.16) and (5.17) of the log marginal likelihood can be decomposed as

$$\mathcal{L}(\boldsymbol{\Theta}) = \gamma_1 + \gamma_2 + \text{const}$$

where we have denoted

$$\gamma_1 = -\frac{1}{2}\log|\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}:}|,$$
$$\gamma_2 = -\frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{U}^{-1}(\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}:})^{-1}\mathbf{U}^{-1}\mathbf{z}_{:}.$$

The vector $z_{:}$ and matrix $\mathbf{U}$ are defined in (5.10) and (5.11), respectively. For the component-wise factorization, change $\mathbf{z}_{:}$ and $\mathbf{U}$ to $\mathbf{z}_d$ and $\mathbf{U}_d$, defined in (5.13) and (5.14), respectively. For the regular GP regression explained in Chapter 4, replace $\mathbf{z}_{:}$ and $\mathbf{U}$ with $\boldsymbol{\Sigma}^{-1}\mathbf{y}$ and $\boldsymbol{\Sigma}^{-1}$, where $\mathbf{y}$ and $\boldsymbol{\Sigma}$ are the observations and the noise covariance matrix, respectively. Also, the covariance matrix $\mathbf{K}_{\mathbf{s}:}$ must be changed appropriately. Note that the inverse $\mathbf{U}^{-1}$ can be evaluated efficiently because the matrix has block-diagonal structure and the inverse preserves the same sparse structure.

We use the following definition:

$$\mathbf{L} = \text{chol}\left(\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}:}\right),$$

where $\text{chol}(\mathbf{X})$ is the lower-triangular matrix in the Cholesky decomposition of $\mathbf{X}$. For a triangular matrix $\mathbf{L}$, $\mathbf{L}^{-1}\mathbf{z}_{:}$ can be evaluated by using forward

or backward substitution, and the determinant $|\mathbf{L}|$ equals the product of the diagonal elements. The computational cost of the Cholesky decomposition is, in general, $O(N^3)$, where N is the dimensionality of the matrix $\mathbf{X}$.

The terms γ_1 and γ_2 can be evaluated as

$$\gamma_1 = -\frac{1}{2}\log|\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_:}| = -\log|\mathbf{L}|$$
$$\gamma_2 = -\frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{U}^{-1}(\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_:})^{-1}\mathbf{U}^{-1}\mathbf{z} = -\frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{U}^{-1}\mathbf{L}^{-\mathrm{T}}\mathbf{L}^{-1}\mathbf{U}^{-1}\mathbf{z} = -\frac{1}{2}\mathbf{b}^{\mathrm{T}}\mathbf{b},$$

where $\mathbf{b} = \mathbf{L}^{-1}\mathbf{U}^{-1}\mathbf{z}_:$. The gradient of γ_1 equals

$$\begin{aligned}\frac{\partial\gamma_1}{\partial\theta_{di}} &= -\frac{1}{2}\frac{\partial}{\partial\theta_{di}}\log|\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_:}| \\ &= -\frac{1}{2}\operatorname{tr}\left(\left(\mathbf{K}_{\mathbf{s}_:} + \mathbf{U}^{-1}\right)^{-1}\frac{\partial\mathbf{K}_{\mathbf{s}_:}}{\partial\theta_{di}}\right), \end{aligned} \tag{B.1}$$

where θ_{di} is the i-th hyperparameter of the d-th latent component $\mathbf{s}_{d:}$. If compactly supported covariance functions are used to produce a sparse covariance matrix $\mathbf{K}_{\mathbf{s}_:}$, one needs to evaluate only those elements of the inverse that correspond to the nonzero elements of $\frac{\partial\mathbf{K}_{\mathbf{s}_:}}{\partial\theta_{di}}$. For evaluating only some elements of the inverse of a sparse matrix, one can utilize, for instance, an efficient implementation by Vanhatalo and Vehtari (2008). The gradient of γ_2 is

$$\begin{aligned}\frac{\partial\gamma_2}{\partial\theta_{di}} &= -\frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{U}^{-1}\frac{\partial}{\partial\theta_{di}}(\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_:})^{-1}\mathbf{U}^{-1}\mathbf{z} \\ &= \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{U}^{-1}(\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_:})^{-1}\frac{\partial\mathbf{K}_{\mathbf{s}_:}}{\partial\theta_{di}}(\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_:})^{-1}\mathbf{U}^{-1}\mathbf{z} \\ &= \frac{1}{2}\mathbf{b}^{\mathrm{T}}\mathbf{L}^{-1}\frac{\partial\mathbf{K}_{\mathbf{s}_:}}{\partial\theta_{di}}\mathbf{L}^{-\mathrm{T}}\mathbf{b} \\ &= \frac{1}{2}\mathbf{c}^{\mathrm{T}}\frac{\partial\mathbf{K}_{\mathbf{s}_:}}{\partial\theta_{di}}\mathbf{c},\end{aligned}$$

where $\mathbf{c} = \mathbf{L}^{-\mathrm{T}}\mathbf{b}$.

The approximate posterior distributions (5.12) and (5.15) equal

$$q(\mathbf{S}) = \mathcal{N}(\mathbf{s}_:|\bar{\mathbf{s}}_:, \mathbf{V}),$$

where

$$\mathbf{V} = \left(\mathbf{K}_{\mathbf{s}_:}^{-1} + \mathbf{U}\right)^{-1} = \mathbf{K}_{\mathbf{s}_:} - \mathbf{K}_{\mathbf{s}_:}\left(\mathbf{K}_{\mathbf{s}_:} + \mathbf{U}^{-1}\right)^{-1}\mathbf{K}_{\mathbf{s}_:}$$
$$\bar{\mathbf{s}}_: = \left(\mathbf{K}_{\mathbf{s}_:}^{-1} + \mathbf{U}\right)^{-1}\mathbf{z}_: = \mathbf{K}_{\mathbf{s}_:}\left(\mathbf{K}_{\mathbf{s}_:} + \mathbf{U}^{-1}\right)^{-1}\mathbf{U}^{-1}\mathbf{z}_: = \mathbf{K}_{\mathbf{s}_:}\mathbf{L}^{-\mathrm{T}}\mathbf{b} = \mathbf{K}_{\mathbf{s}_:}\mathbf{c}.$$

The covariance matrix $\mathbf{V}$ is, in general, a full matrix even if $\mathbf{K}_{\mathbf{s}_:}$ is sparse. Therefore, one may want to avoid evaluating the full covariance matrix. Note

that the VB-EM algorithm for GPFA does not need the full covariance matrix as the algorithm only uses a small portion of the covariance matrix in the terms $\left\langle \mathbf{s}_{:n}\mathbf{s}_{:n}^{\mathrm{T}} \right\rangle$. The log determinant of the covariance matrix can be evaluated as

$$\begin{aligned}\log|\mathbf{V}| &= \left|\mathbf{K}_{\mathbf{s}_:}^{-1} + \mathbf{U}\right| \\ &= \log\left|\mathbf{K}_{\mathbf{s}_:}\left(\mathbf{K}_{\mathbf{s}_:} + \mathbf{U}^{-1}\right)^{-1}\mathbf{U}^{-1}\right| \\ &= 2\log|\mathbf{L}_{\mathbf{K}}| - 2\log|\mathbf{L}| - 2\log|\mathbf{L}_{\mathbf{U}}|,\end{aligned}$$

where $\mathbf{L}_{\mathbf{K}} = \mathrm{chol}(\mathbf{K}_{\mathbf{s}_:})$ and $\mathbf{L}_{\mathbf{U}} = \mathrm{chol}(\mathbf{U})$, which are efficient to evaluate because $\mathbf{K}_{\mathbf{s}_:}$ is block-diagonal and $\mathbf{U}$ is sparse.

B.2 Sparse approximation

This section gives details on implementing the variational sparse approximation presented in Section 5.3. We give formulas for evaluating the lower bounds, their gradients and the approximate posterior distribution $q(\hat{\mathbf{S}})$.

The lower bounds (5.21) and (5.22) of the log marginal likelihood can be decomposed as

$$\mathcal{L}\left(\hat{\mathrm{T}}, \boldsymbol{\Theta}\right) = \gamma_1 + \gamma_2 + \gamma_3 + \mathrm{const},$$

where we have denoted

$$\begin{aligned}\gamma_1 &= -\frac{1}{2}\log\left|\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\right|, \\ \gamma_2 &= -\frac{1}{2}\mathbf{z}_:^{\mathrm{T}}\mathbf{U}^{-1}\left(\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\right)^{-1}\mathbf{U}^{-1}\mathbf{z}_:, \\ \gamma_3 &= -\frac{1}{2}\mathrm{tr}\left[(\mathbf{K}_{\mathbf{s}_:} - \mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:})\mathbf{U}\right].\end{aligned}$$

The vector $z_:$ and matrix $\mathbf{U}$ are defined in (5.10) and (5.11), respectively. For the component-wise factorization, change $\mathbf{z}_:$ and $\mathbf{U}$ to $\mathbf{z}_d$ and $\mathbf{U}_d$, defined in (5.13) and (5.14), respectively. For the regular GP regression explained in Chapter 4, replace $\mathbf{z}_:$ and $\mathbf{U}$ with $\boldsymbol{\Sigma}^{-1}\mathbf{y}$ and $\boldsymbol{\Sigma}^{-1}$, where $\mathbf{y}$ and $\boldsymbol{\Sigma}$ are the observations and the noise covariance matrix, respectively. Also, the covariance matrices $\mathbf{K}_{\mathbf{s}_:}$, $\mathbf{K}_{\hat{\mathbf{s}}_:}$, $\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}$ and $\mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}$ must be changed appropriately. Note that the inverse $\mathbf{U}^{-1}$ can be evaluated efficiently because the matrix has block-diagonal structure and the inverse preserves the same sparse structure.

In the formulas, we use the following definitions:

$$\begin{aligned}\mathbf{L}_{\mathbf{K}} &= \mathrm{chol}(\mathbf{K}_{\hat{\mathbf{s}}_:}), \\ \boldsymbol{\Lambda} &= \mathbf{K}_{\hat{\mathbf{s}}_:} + \mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\mathbf{U}\mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:},\end{aligned}$$

where chol($\mathbf{X}$) is the lower-triangular matrix in the Cholesky decomposition of matrix $\mathbf{X}$, as discussed in the previous section.

The terms γ_1, γ_2 and γ_3 can be evaluated as follows. The first term γ_1 equals

$$\begin{aligned}
\gamma_1 &= -\frac{1}{2}\log\left|\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\right| \\
&= -\frac{1}{2}\log\left|\mathbf{I} + \mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\mathbf{U}\mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}\right| + \text{const} \\
&= \frac{1}{2}\log|\mathbf{K}_{\hat{\mathbf{s}}_:}| - \frac{1}{2}\log|\mathbf{K}_{\hat{\mathbf{s}}_:} + \mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\mathbf{U}\mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}| + \text{const} \\
&= \frac{1}{2}\log|\mathbf{K}_{\hat{\mathbf{s}}_:}| - \frac{1}{2}\log|\mathbf{\Lambda}| + \text{const},
\end{aligned}$$

where the second line is obtained by using the matrix determinant lemma (A.3). The second term γ_2 equals

$$\begin{aligned}
\gamma_2 &= -\frac{1}{2}\mathbf{z}_:^{\mathrm{T}}\mathbf{U}^{-1}\left(\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\right)^{-1}\mathbf{U}^{-1}\mathbf{z}_: \\
&= -\frac{1}{2}\mathbf{z}_:^{\mathrm{T}}\mathbf{U}^{-1}\left(\mathbf{U} - \mathbf{U}\mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}\mathbf{\Lambda}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\mathbf{U}\right)\mathbf{U}^{-1}\mathbf{z}_: \\
&= -\frac{1}{2}\mathbf{z}_:^{\mathrm{T}}\left(\mathbf{U}^{-1} - \mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}\mathbf{\Lambda}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\right)\mathbf{z}_:,
\end{aligned}$$

where the second line is obtained by using the matrix inversion lemma (A.1). The third term γ_3 equals

$$\begin{aligned}
\gamma_3 &= -\frac{1}{2}\operatorname{tr}\left[(\mathbf{K}_{\mathbf{s}_:} - \mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:})\mathbf{U}\right] \\
&= -\frac{1}{2}\sum_{n=1}^{N}\sum_{d=1}^{D}(k_{s_{dn}} - \mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}}^{\mathrm{T}}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}})\left[\mathbf{U}_n\right]_{dd},
\end{aligned}$$

because $\mathbf{U}$ is block-diagonal with respect to $n = 1, \ldots, N$ and the GP covariance matrices are block-diagonal with respect to $d = 1, \ldots, D$. Thus, they have overlapping nonzero elements on the main diagonal only.

The gradients can be evaluated by using the basic rules of matrix derivatives, including the rules for derivatives of the logarithm of a determinant (A.4) and the inverse of a matrix (A.5). The gradient of the first term γ_1 can be

derived as

$$\begin{aligned}
\frac{\partial \gamma_1}{\partial \theta_{di}} =& -\frac{1}{2}\frac{\partial}{\partial \theta_{di}} \log \left|\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\right| \\
=& -\frac{1}{2} \operatorname{tr}\left[\left(\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\right)^{-1} \frac{\partial}{\partial \theta_{di}} \left(\mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\right)\right] \\
=& \frac{1}{2} \operatorname{tr}\left[\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}} \left(\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\right)^{-1} \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_{:}}}{\partial \theta_{di}}\right] - \\
& \operatorname{tr}\left[\left(\mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\right)^{-1} \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}}{\partial \theta_{di}}\right] \\
=& \frac{1}{2} \operatorname{tr}\left[\left(\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} - \left(\mathbf{K}_{\hat{\mathbf{s}}_{:}} + \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}} \mathbf{U} \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}}\right)^{-1}\right) \frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_{:}}}{\partial \theta_{di}}\right] - \\
& \operatorname{tr}\left[\mathbf{U} \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \left(\mathbf{K}_{\hat{\mathbf{s}}_{:}} + \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}} \mathbf{U} \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}}\right)^{-1} \frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}}{\partial \theta_{di}}\right] \\
=& \frac{1}{2} \operatorname{tr}\left[\left(\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} - \mathbf{\Lambda}^{-1}\right) \frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_{:}}}{\partial \theta_{di}}\right] - \operatorname{tr}\left[\mathbf{U} \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{\Lambda}^{-1} \frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}}{\partial \theta_{di}}\right],
\end{aligned}$$

where the fourth line is obtained by using the matrix inversion lemmas (A.1) and (A.2), and θ_{di} is the i-th hyperparameter of the d-th latent component $\mathbf{s}_{d:}$. By denoting $\mathbf{\Psi} = \mathbf{U}^{-1} + \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}$, the gradient of the second term γ_2 is

$$\begin{aligned}
\frac{\partial \gamma_2}{\partial \theta_{di}} =& -\frac{1}{2} \mathbf{z}_{:}^{\mathrm{T}} \mathbf{U}^{-1} \frac{\partial \mathbf{\Psi}^{-1}}{\partial \theta_{di}} \mathbf{U}^{-1} \mathbf{z}_{:} \\
=& \frac{1}{2} \mathbf{z}_{:}^{\mathrm{T}} \mathbf{U}^{-1} \mathbf{\Psi}^{-1} \frac{\partial}{\partial \theta_{di}} \left(\mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\right) \mathbf{\Psi}^{-1} \mathbf{U}^{-1} \mathbf{z}_{:} \\
=& -\frac{1}{2} \mathbf{z}_{:}^{\mathrm{T}} \mathbf{U}^{-1} \mathbf{\Psi}^{-1} \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_{:}}}{\partial \theta_{di}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}} \mathbf{\Psi}^{-1} \mathbf{U}^{-1} \mathbf{z}_{:} + \\
& \mathbf{z}_{:}^{\mathrm{T}} \mathbf{U}^{-1} \mathbf{\Psi}^{-1} \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1} \frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}}{\partial \theta_{di}} \mathbf{\Psi}^{-1} \mathbf{U}^{-1} \mathbf{z}_{:} \\
=& -\frac{1}{2} \mathbf{z}_{:}^{\mathrm{T}} \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{\Lambda}^{-1} \frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_{:}}}{\partial \theta_{di}} \mathbf{\Lambda}^{-1} \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}} \mathbf{z}_{:} + \\
& \mathbf{z}_{:}^{\mathrm{T}} \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{\Lambda}^{-1} \frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}}{\partial \theta_{di}} \left(\mathbf{I} - \mathbf{U} \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}} \mathbf{\Lambda}^{-1} \mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\right) \mathbf{z}_{:},
\end{aligned}$$

where the last line is obtained by using the matrix inversion lemma (A.2). The

gradient of the third term γ_3 equals

$$\begin{aligned}\frac{\partial \gamma_3}{\partial \theta_{di}} =& -\frac{1}{2}\sum_{n=1}^{N}\sum_{d=1}^{D}\frac{\partial}{\partial \theta_{di}}\left(k_{s_{dn}} - \mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}}^{\mathrm{T}}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}}\right)[\mathbf{U}_n]_{dd} \\ =& -\frac{1}{2}\sum_{n=1}^{N}\sum_{d=1}^{D}\left(\frac{\partial k_{s_{dn}}}{\partial \theta_{di}} + \mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}}^{\mathrm{T}}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_:}}{\partial \theta_{di}}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}} - \right. \\ & \left. 2\mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}}^{\mathrm{T}}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\frac{\partial \mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}}}{\partial \theta_{di}}\right)[\mathbf{U}_n]_{dd}\,.\end{aligned}$$

In order to evaluate the terms, we need the following variables:

$$\begin{aligned}\mathbf{L}_{\mathbf{K}} &= \mathrm{chol}(\mathbf{K}_{\hat{\mathbf{s}}_:}),\\ \mathbf{L}_{\mathbf{\Lambda}} &= \mathrm{chol}(\mathbf{\Lambda}),\\ \mathbf{L}_{\mathbf{U}} &= \mathrm{chol}(\mathbf{U}),\\ \mathbf{R} &= \mathbf{L}_{\mathbf{\Lambda}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\\ \boldsymbol{\zeta} &= \mathbf{L}_{\mathbf{\Lambda}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\mathbf{z}_: = \mathbf{R}\mathbf{z}_:,\\ \boldsymbol{\nu} &= \mathbf{\Lambda}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}\mathbf{z}_: = \mathbf{L}_{\mathbf{\Lambda}}^{-\mathrm{T}}\boldsymbol{\zeta},\end{aligned}$$

where $\mathbf{R}$ dominates the cost. Exploiting the block structure of $\mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}$, the computational cost is $O\left(N\left(\sum_{d=1}^{D}\hat{N}_d\right)^2\right)$, where $\hat{N}_d$ is the number of inducing inputs for the d-th component and N is the number of inputs. Using the above definitions, the terms can be evaluated as

$$\begin{aligned}\gamma_1 =& \log|\mathbf{L}_{\mathbf{K}}| - \log|\mathbf{L}_{\mathbf{\Lambda}}| + \mathrm{const},\\ \gamma_2 =& -\frac{1}{2}(\mathbf{L}_{\mathbf{U}}^{-1}\mathbf{z}_:)^{\mathrm{T}}(\mathbf{L}_{\mathbf{U}}^{-1}\mathbf{z}_:) + \frac{1}{2}\boldsymbol{\zeta}^{\mathrm{T}}\boldsymbol{\zeta},\\ \gamma_3 =& -\frac{1}{2}\sum_{n=1}^{N}\sum_{d=1}^{D}\left[k_{s_{dn}} - (\mathbf{L}_{\mathbf{K}}^{-1}\mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}})^{\mathrm{T}}(\mathbf{L}_{\mathbf{K}}^{-1}\mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}})\right][\mathbf{U}_n]_{dd}\,,\\ \frac{\partial \gamma_1}{\partial \theta_{di}} =& \frac{1}{2}\,\mathrm{tr}\left[\left(\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1} - \mathbf{\Lambda}^{-1}\right)\frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_:}}{\partial \theta_{di}}\right] - \mathrm{tr}\left[\mathbf{U}\mathbf{R}^{\mathrm{T}}\mathbf{L}_{\mathbf{\Lambda}}^{-1}\frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}}{\partial \theta_{di}}\right],\\ \frac{\partial \gamma_2}{\partial \theta_{di}} =& -\frac{1}{2}\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_:}}{\partial \theta_{di}}\boldsymbol{\nu} + \boldsymbol{\nu}^{\mathrm{T}}\frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_:,\mathbf{s}_:}}{\partial \theta_{di}}(\mathbf{z}_: - \mathbf{U}\mathbf{K}_{\mathbf{s}_:,\hat{\mathbf{s}}_:}\boldsymbol{\nu}),\\ \frac{\partial \gamma_3}{\partial \theta_{di}} =& -\frac{1}{2}\sum_{n=1}^{N}\sum_{d=1}^{D}\left(\frac{\partial k_{s_{dn}}}{\partial \theta_{di}} + \mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}}^{\mathrm{T}}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\frac{\partial \mathbf{K}_{\hat{\mathbf{s}}_:}}{\partial \theta_{di}}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}} - \right.\\ & \left. 2\mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}}^{\mathrm{T}}\mathbf{K}_{\hat{\mathbf{s}}_:}^{-1}\frac{\partial \mathbf{k}_{\hat{\mathbf{s}}_{d:},s_{dn}}}{\partial \theta_{di}}\right)[\mathbf{U}_n]_{dd}\,.\end{aligned}$$

Note that in most terms one can utilize the block structure of $\mathbf{K}_{\hat{\mathbf{s}}_{:}}$, $\mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}$ and their gradients in order to reduce the computational cost. For instance, the latter trace term in $\frac{\partial \gamma_1}{\partial \theta_{di}}$ can be evaluated with a cost $O\left(N\left(\sum_{d=1}^{D}\hat{N}_d\right)^2\right)$.

Using the above variable definitions, the approximate posterior distributions (5.19) and (5.20) can be evaluated as

$$q(\hat{\mathbf{S}}) = \mathcal{N}\left(\hat{\mathbf{s}}_{:} \left| \bar{\hat{\mathbf{s}}}_{:}, \mathbf{V}\right.\right),$$

where the parameters are defined as

$$\begin{aligned}\bar{\hat{\mathbf{s}}}_{:} &= \mathbf{K}_{\hat{\mathbf{s}}_{:}}\mathbf{L}_{\mathbf{\Lambda}}^{-\mathrm{T}}\mathbf{R}\mathbf{z}_{:}\\ \mathbf{V} &= \mathbf{K}_{\hat{\mathbf{s}}_{:}}\mathbf{\Lambda}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{:}}.\end{aligned}$$

The posterior predictive distribution $q(\mathbf{S}) = \int p(\mathbf{S}|\hat{\mathbf{S}})q(\hat{\mathbf{S}})d\hat{\mathbf{S}}$ equals

$$q(\mathbf{S}) = \mathcal{N}\left(\mathbf{s}_{:} \left| \bar{\mathbf{s}}_{:}, \mathbf{V}_{\mathbf{s}_{:}}\right.\right)$$

where the mean is defined as

$$\bar{\mathbf{s}}_{:} = \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}}\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1}\left\langle\hat{\mathbf{s}}_{:}\right\rangle = \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}}\mathbf{L}_{\mathbf{\Lambda}}^{-\mathrm{T}}\mathbf{R}\mathbf{z}_{:} = \mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{z}_{:} = \mathbf{R}^{\mathrm{T}}\boldsymbol{\zeta},$$

and the covariance as

$$\begin{aligned}\mathbf{V}_{\mathbf{s}_{:}} &= \mathbf{K}_{\mathbf{s}_{:}} - \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}}\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}} + \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}}\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1}\mathbf{V}\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\\ &= \mathbf{K}_{\mathbf{s}_{:}} - \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}}\mathbf{K}_{\hat{\mathbf{s}}_{:}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}} + \mathbf{K}_{\mathbf{s}_{:},\hat{\mathbf{s}}_{:}}\mathbf{\Lambda}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}\\ &= \mathbf{K}_{\mathbf{s}_{:}} - (\mathbf{L}_{\mathbf{K}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}})^{\mathrm{T}}(\mathbf{L}_{\mathbf{K}}^{-1}\mathbf{K}_{\hat{\mathbf{s}}_{:},\mathbf{s}_{:}}) + \mathbf{R}^{\mathrm{T}}\mathbf{R}.\end{aligned}$$

Note that the learning algorithm does not need the full covariance matrix $\mathbf{V}_{\mathbf{s}_{:}}$ but only small blocks of it for the terms $\left\langle\mathbf{s}_{:n}\mathbf{s}_{:n}^{\mathrm{T}}\right\rangle$.