# Gaussian Process Models
for Robust Regression, Classification, and Reinforcement Learning

Vorgelegt von
Diplom Informatiker Malte Kuß
aus Wolfsburg

März 2006

Genehmigte Dissertation zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat.)
am Fachbereich Informatik der Technischen Universität Darmstadt
(Hochschulkennziffer D17)

Erstreferent:	Prof. Dr. Thomas Hofmann	Eingereicht am	13. Februar 2006
Korreferenten:	PhD. Carl E. Rasmussen	Tag der Disputatiton	21. März 2006
	Prof. Dr. Bernt Schiele		

Erklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit—mit Ausnahme der in ihr ausdrücklich genannten Hilfen—selbständig verfasst habe.

Wissenschaftlicher Werdegang

10/96 – 02/02 Studium der Informatik an der Technischen Universität Berlin

- Nebenfach Wirtschaftswissenschaften (VWL)
- Studienschwerpunkte: Statistik, Maschinelles Lernen, Softwaretechnik, Datenbanken, Mikroökonomie, Spieltheorie
- Diplomarbeit am Lehrstuhl für Wirtschaftsmathematik und Statistik zum Thema „Non-linear Multivariate Analysis with Geodesic Kernels" (Prof. Kockelkorn)
- Diplom mit Auszeichnung

07/02 – 03/06 Doktorand am Max-Planck-Institut für biologische Kybernetik, Tübingen

- Arbeitsgruppe für empirische Inferenz (Prof. Schölkopf)
- Forschungsinteressen: Bayesianische Statistik, Entscheidungstheorie, Monte Carlo Methoden
- Promotion an der Technischen Universität Darmstadt (Prof. Hofmann)

Referenz

```
@PhdThesis{Kuss:06,
  author =  {M. Kuss},
  title =  {Gaussian Process Models for Robust Regression,
                  Classification, and Reinforcement Learning},
  school =  {Technische Universit{\"a}t Darmstadt},
  year =  {2006}
}
```

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit Erweiterungen und Anwendungen einer Klasse von statistischen Modellen, den so genannten Gauß-Prozess Modellen. Methoden des überwachten Lernens, wie sie z.B. in der Regressions- und Diskriminanzanalyse verwendet werden, zielen darauf ab, Abhängigkeiten zwischen Variablen zu identifizieren und das so gewonnene Verständnis über den datengenerierenden Prozess zur Vorhersage zu nutzen. Die in dieser Arbeit untersuchten Modelle beruhen auf der Annahme, dass diese Abhängigkeiten in einen systematischen Zusammenhang und eine zufällige Komponente zerlegt werden können, wobei die systematische Zusammenhang mittels einer latenten Funktion beschrieben werden kann. Als Gauß-Prozess Modelle bezeichnet man statistische Modelle, in denen ein Gauß-Prozess verwendet wird, um die Bayesianische a priori Unsicherheit über diese latente Funktion zu beschreiben.

Nach einer kurzen Einführung in die Bayesianische Statistik in Kapitel 2 wird in Kapitel 3 die Klasse der Gauß-Process Modelle detailliert beschrieben. Darüber hinaus wird darauf eingegangen, wie der Gauß-Prozess zur Beschreibung der a priori Unsicherheit verstanden werden kann.

Der konzeptionellen Klarheit des Bayesianischen Ansatzes stehen oftmals praktische Schwierigkeiten gegenüber, da die auftretenden Integrale nicht analytisch lösbar sind. Approximationstechniken sind daher von zentraler Bedeutung für die Anwendung Bayesianischer Methoden in der praktischen Datenanalyse. In Kapitel 4 werden Laplaces Methode, Expectation Propagation und Markov chain Monte Carlo Verfahren beschrieben sowie deren Anwendung in Gauß-Prozess Modellen.

Unter den Gauß-Prozess Modellen sticht das Regressionmodell mit normalverteilter Störgröße heraus, da unter diesen Annahmen Bayesianische Inferenz analytisch handhabbar ist und die a posteriori Unsicherheit über die latente Funktion ebenfalls durch einen Gauß-Process beschrieben werden kann. Allerdings macht die Annahme der Normalverteilung das Modell sensitiv gegenüber Ausreissern, d.h. Beobachtungen die stark von der systematischen Struktur abweichen. Kapitel 5 beschreibt verschiedene Gauß-Prozess Modelle für nichtlineare robuste Regressionsanalyse. In diesen robusten Regressionsmodellen wird die Verteilung der Störgröße durch eine leptokurtotische (heavy-tailed) Verteilungen beschrieben.

Kapitel 6 beschäftigt sich mit dem Gauß-Prozess Modell zur binären Klassifikationsanalyse. In der Literatur finden sich verschiedene Ansätze, wie man Bayesianische Inferenz in diesem Modell approximieren kann. Allerdings bestand bisher Unklarheit darüber wie akkurat diese Näherungsverfahren sind und welches in der Praxis zu bevorzugen ist. Dieses Fragen werden sowohl theoretisch durch eine Betrachtung der Struktur der a posteriori Verteilung als auch experimentell durch einen Vergleich mit aufwendigen Markov chain Monte Carlo Simulationen beantwortet.

Als Reinforcement Lernen bezeichnet man die das adaptive Lernen in sequentiellen Entscheidungsproblemen. Kapitel 7 beschreibt Anwendungen von Gauß-Prozess Regressionsmodellen für Reinforcement Lernen in Problem mit kontinuierlichen Zustandsräumen. Dabei werden verschiedene Möglichkeiten vorgestellt wie man Gauss-Prozesse nutzen kann, um die Effekte der Entscheidungen vorherzusagen und um die so genannte Value Funktion zu repräsentieren.

Summary

Gaussian process models constitute a class of probabilistic statistical models in which a Gaussian process (GP) is used to describe the Bayesian *a priori* uncertainty about a latent function.

After a brief introduction of Bayesian analysis, Chapter 3 describes the general construction of GP models with the conjugate model for regression as a special case (O'Hagan, 1978). Furthermore, it will be discussed how GP can be interpreted as priors over functions and what beliefs are implicitly represented by this.

The conceptual clearness of the Bayesian approach is often in contrast with the practical difficulties that result from its analytically intractable computations. Therefore approximation techniques are of central importance for applied Bayesian analysis. Chapter 4 describes Laplace's method, the Expectation Propagation approximation, and Markov chain Monte Carlo sampling for approximate inference in GP models.

The most common and successful application of GP models is in regression problems where the noise is assumed to be homoscedastic and distributed according to a normal distribution. In practical data analysis this assumption is often inappropriate and inference is sensitive to the occurrence of more extreme errors (so called outliers). Chapter 5 proposes several variants of GP models for robust regression and describes how Bayesian inference can be approximated in each. Experiments on several data sets are presented in which the proposed models are compared with respect to their predictive performance and practical applicability.

Gaussian process priors can also be used to define flexible, probabilistic classification models. Again, exact Bayesian inference is analytically intractable and various approximation techniques have been proposed, but no clear picture has yet emerged, as to when and why which algorithm should be preferred. Chapter 6 presents a detailed examination of the model, focusing on the question which approximation technique is most appropriate by investigating the structure of the posterior distribution. An experimental study is presented which corroborates the theoretical insights.

Reinforcement learning deals with the problem of how an agent can optimise its behaviour in a sequential decision process such that its utility over time is maximised. Chapter 7 addresses applications of GPs for model-based reinforcement learning in continuous domains. If the environment's response to the agent's actions can be predicted using GP regression models, probabilistic planning and an approximate policy iteration algorithm can be implemented. A core concept in reinforcement learning is the value function, which describes the long-term strategic value of a state. Using GP models we are able to solve an approximate continuous equivalent of the Bellman equations, and it will be shown how this can be used to estimate value functions.

Contents

Acknowledgements

This thesis is a product of my work in the department of empirical inference for machine learning and perception at the Max Planck Institute for Biological Cybernetics in Tübingen, Germany. At first I would like to thank Bernhard Schölkopf for letting me be part of his group and for providing an excellent environment for research. In this regard I also would like to express my gratitude to Sabrina Nielebock and Sebastian Stark.

After arriving in Tübingen I had the fortune to meet Carl Edward Rasmussen who became my advisor and from whom I learned a lot about Gaussian processes and many other things. I would like to thank him for his generosity in sharing his thoughts and knowledge.

Moreover, I wish to thank the members of my committee at the Technische Universität Darmstadt, particularly my official "Doktorvater" Thomas Hofmann for his insightful comments on an early draft.

My interest in machine learning was sparked years ago by Thore Graepel and Ralf Herbrich back at the Technische Universität Berlin. A posteriori, I would like to thank them for the encouragement to go to Tübingen and later for inviting me to spend several weeks with them in Cambridge.

In the course of my study of Bayesian inference I had the pleasure to work with and learn from many researchers in Tübingen and elsewhere. My understanding of Gaussian process models greatly benefited from discussions and collaborations with Tobias Pfingsten, Joaquin Quiñonero Candela, Anton Schwaighofer, Matthias Seeger, and my former office mate Lehel Csató. Furthermore, I would like to thank Christine & Felix Wichmann, Jeremy Hill, HyunJung Shin, Ule von Luxburg, Dilan Görür, Thomas Navin Lal, Matthias Hein, Jan Eichhorn, Olivier Chapelle, and Peter Gehler for their advise, encouragement, and friendship.

In the days before I submitted this thesis Jeremy Hill, Alexander Zien, Matthias Hein, Tobias Pfingsten, and especially Dilan Görür helped me a lot by proof-reading parts of (or even the whole) manuscript. Of course, any mistakes yet uncovered remain mine.

Finally, this is also the place to thank my parents and their spouses for their generous support and encouragement during my studies and before.

The most beautiful thing during the past years was to have Claudia Beckmann by my side. Doing research and writing a thesis involves moments of doubt and frustration. But she always knew how to cheer me up and has proven an enormous patience with me in times when I lost the necessary distance to my work.

Symbols & Abbreviations

General

GP	Gaussian process (defined in Section 3.3.1 on page 25)
MCMC	Markov chain Monte Carlo
AIS	Annealed Importance Sampling
HMC	Hybrid Monte Carlo
EP	Expectation Propagation
GPC	Gaussian Process Classification
SVM	Support Vector Machine (classification)
SVR	Support Vector Regression
ARD	Automatic relevance determination (see Section 3.3.4 on page 30)
ln	Natural logarithm (base e)
$\approx$	The left hand side is approximated by the right hand side expression
$\leftarrow$	The left hand side is assigned the value of the right hand side expression
p.s.d.	Positive semi-definite
∇f	The gradient of f
$\nabla\nabla f$	The Hessian matrix of second derivatives of f

Matrix Analysis

$\mathbf{a}, \mathbf{b}, \mathbf{c}$	Column vectors (bold lower case letters)
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	Matrices (bold capital letters)
$\mathbf{a}_i$	The ith vector $\mathbf{a}$
a_i	The ith element of vector $\mathbf{a}$
A_{ij}	The jth element in the ith row of matrix $\mathbf{A}$
$\mathbf{A}^\top$	Transpose of a matrix
$\mathrm{tr}(\mathbf{A})$	Trace of a square matrix
$\lvert\mathbf{A}\rvert$	Determinant
$\mathbf{A}^{-1}$	Inverse of a square matrix
$\mathbf{A}^{+}$	Pseudoinverse (see Appendix B.1.2 on page 168)
$\mathrm{diag}(\mathbf{a})$	A diagonal matrix with diagonal entries $\mathbf{a}$
$\mathbf{1}$	Vector of ones
$\mathbf{I}$	The identity matrix $\mathbf{I} = \mathrm{diag}(\mathbf{1})$
$\mathbf{e}_i$	Unit vector in ith dimension, the ith column of $\mathbf{I}$
$[\mathbf{A}\,\mathbf{B}]$	Square brackets denote partitioned matrices

Probability

No notational distinction is made between probabilities and probability density functions. The parameterisation of probability distributions follows Gelman et al. (1995, Appendix A).

$p(\mathbf{x})$	A probability (density) function
$q(\mathbf{x})$	An approximation to $p(\mathbf{x})$
cov	Covariance of random variables
var	Variance of a random variable
$y\|x$	Conditional random variable
$\boldsymbol{\phi}$	Parameters of a model
$\boldsymbol{\psi}$	Hyper-parameters, parameters of a hyper-prior in a hierarchical prior construction
$\mathcal{D}$	Data, all observed samples
$\mathcal{M}$	Model, a hypothesis about the data-generating process
L	Loss function
R	Risk, expected loss (see definitions in Section 2.2 on page 8)
$\mathcal{N}(\mathbf{x}\|\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate normal distribution (see Appendix B.2 on page 170)
$\mathcal{T}(x\|\mu, \sigma^2, \nu)$	Student-t distribution (see description in Section 5.19 on page 68)
KL	Kullback-Leibler divergence (defined in Appendix B.2.3.6 on page 172)
$\mathcal{H}$	Entropy of a distribution (defined in Appendix B.2.3.5 on page 171)
MAP	Maximum *a posteriori* estimate (defined in Section 2.2 on page 8)
p.d.f.	Probability density function
c.d.f.	Cumulative distribution function
i.i.d.	Independent and identically distributed

Gaussian process specific notation

$\mathcal{X}$	Input space, index set of a Gaussian process
m	Number of observed examples, the size of $\mathcal{D}$
n	Number of input dimensions
$\mathbf{X}$	An $[m \times n]$ matrix of observed inputs (explanatory variables)
$\mathbf{y}$	An $[m \times 1]$ vector of target values (responses, labels)
f	A (latent) function, or a Gaussian process
$\mathbf{f}$	An $[m \times 1]$ vector $[f(\mathbf{x}_1), \dots, f(\mathbf{x}_m)]^\top$
$m(\mathbf{x})$	Mean function of a Gaussian process $m : \mathcal{X} \to \mathbb{R}$
$k(\mathbf{x}, \mathbf{x}')$	Covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, parameterised by $\boldsymbol{\psi}$
$\mathbf{K}$	An $[m \times m]$ covariance matrix, where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$
$\mathbf{k}(\mathbf{x})$	An $[m \times 1]$ vector of covariances $[k(\mathbf{x}_1, \mathbf{x}), \dots, k(\mathbf{x}_m, \mathbf{x})]^\top$
$\boldsymbol{\theta}$	Additional parameters of the likelihood $p(y\|f(\mathbf{x}), \boldsymbol{\theta})$ besides f
ℓ	Characteristic length scale (described in Section 3.3.4 on page 30)
$\mathcal{F}$	Feature space (see Section 3.4.1 on page 34)
σ_s^2	Signal variance (introduced in Section 3.3.2 on page 27)
σ_n^2	Noise variance in the conjugate model (see Section 3.2 on page 18)

Reinforcement learning notation

MDP	Markov decision process
$\boldsymbol{s} \in \mathcal{S}$	State from the set of all possible states
$a \in \mathcal{A}(\boldsymbol{s})$	Action from the set of all available actions in a given state
r	Reward, can be a random variable
r	Expected reward, $\mathbb{E}[r]$
$\mathfrak{p} \in \mathfrak{P}$	Policy (or strategy) from the set of all policies
V	The state value function $\mathsf{V} : \mathcal{S} \to \mathbb{R}$ (see Section 7.1.2 on page 123)
$\mathcal{Q}$	The state-action-value function $\mathcal{Q} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ (see Section 7.1.2)
γ	Discount rate

1. Introduction

The practical relevance of a theory is seen in its ability to solve problems, which is usually synonymous to help making good decisions. Decision making is based on hypothesising and learning about the *state of nature*, given limited experience and incomplete information. Given that the ability to learn from examples is fundamental to many aspects of life it clearly stands out as a setting that is worth studying.

Learning, in the sense as it will be used below, is an inductive process in which the reasoning is from the particular to the general. The task is to make inference, to draw conclusions about the population based on empirical observations. Based on finitely many samples and other available information a model, i.e. a mathematical description, is sought that captures the dependencies between observed quantities. Subsequently this structure can be used to make statements about the underlying data-generating process. In particular, the dependencies thus identified can be used to derive rules that allow prediction of properties of yet unobserved instances.

Historically the formal study and development of prediction techniques was part of statistical science. Classical models in multivariate analysis often assume normality of the population distribution and linear dependencies. The applicability of linear models is limited because of these rigid assumptions which are often not met in practical data analysis problems. With the advent of modern computers more data could be handled and more complex and flexible models were developed, e.g. generalised linear models and non-parametric techniques. Furthermore, the observation that biological systems implement adaptive behaviour motivated the study and simulation of artificial neural networks. It was realised that these models constitute a general class of function approximators, applicable for prediction in various applications. Finally, *machine learning* evolved as a discipline bundling the efforts towards understanding, applying, and improving predictive models that had been developed in statistics, computer science, and engineering.

The analysis of complex systems, e.g. as they appear in bioinformatics or econometrics, often demands flexible nonlinear models in order to capture the dependencies in the data accurately. With the development of increasingly complex models several fundamental problems became apparent. Among the most urgent questions are how to choose an appropriate type of model, how to adapt the model in view of the observed data, e.g. parameter estimation, and how to judge the reliability of the model's predictions.

One particular issue common to almost all data modelling is the question of how flexible a model is allowed to be, i.e. how much complexity the model is able to capture. The optimal flexibility of a model must be in some sense related to the amount of information contained in the observations. If the adaptation of the model is not constrained, nothing prevents a very flexible model from pure memorisation of the examples without capturing—learning—the underlying structures. Furthermore, the increase in flexibility and predictive performance often comes at the cost of a loss in interpretability of the model.

Historically, maximum likelihood estimation has often been the conventional way of parameter estimation in mathematical statistics and has also been used, e.g., in back-propagation training of artificial neural networks. But the more flexible a model is the more this approach is prone to over-fitting, i.e. the misinterpretation of random fluctuation in the data as systematic structure. The practical problem is to set the tradeoff between the accuracy of the model in explaining the observations and the complexity of the structure used for this. Various heuristic approaches had been proposed until the problem was studied systematically in *statistical learning theory*, leading to the concept of structural risk minimisation (Vapnik, 1998, 1999). In this context a new category of classification and regression algorithms named Support Vector Machines was developed, mating the idea of maximum margin separation and kernel based algorithms.

Another approach to the problem of handling complex statistical models was taken by recalling the Bayesian tradition. The Bayesian school of statistics is based on a different notion of probability, by which the probability of an event equals the subjective degree of belief in its occurrence. Together with the conventional rules of mathematical probability this constitutes a framework for inference in probabilistic models, i.e. constructions describing dependencies between entities in terms of probabilities. Inference is performed according to Bayes' rule which describes how beliefs about the data-generating process have to be changed to consistently incorporate the information contained in observed examples. The work presented in this thesis follows the Bayesian approach which will be described in more detail in the following chapter.

MacKay (1992a) and Neal (1996) analysed artificial neural networks using Bayesian techniques which allowed them to control the effective flexibility of the model so that large neural networks could be handled. In this context Gaussian process models were recognised as the limit of a Bayesian neural network as the number of hidden units goes to infinity. This observation can be linked to earlier work by Blight and Ott (1975) and O'Hagan (1978) who introduced the use of Gaussian processes as priors over functions in the statistics literature.

Gaussian process models and kernel methods, like the popular Support Vector Machines, are closely related in the sense that positive definite functions play an important rôle in both. In Support Vector Machines kernel functions are interpreted geometrically, as an inner product in a nonlinearly related feature space. In Gaussian process models the same class of functions is used to describe the covariance of latent function values. Therefore, Gaussian process models have also been referred to as Bayesian kernel machines.

This thesis presents extensions and applications of Gaussian process models from a Bayesian point of view. The main contributions are several models for robust regression, a study of approximation techniques for binary classification, and applications in reinforcement learning. Throughout, the understanding and numerically stable implementation of approximation techniques is an important concern, being a prerequisite for future use in practical data analysis. In the course of this work, a wide range of machine learning problems are addressed, demonstrating the versatility of Gaussian process models.

After a brief introduction of Bayesian analysis, Chapter 3 describes the general construction of Gaussian process models with the conjugate model for regression as a special case. Furthermore, it will be discussed how Gaussian processes can be interpreted as priors over functions and what beliefs are implicitly represented by this.

The conceptual clearness of the Bayesian approach is often in contrast with the practical difficulties that result from its analytically intractable computations. Therefore approximation techniques are of central importance for applied Bayesian analysis. Chapter 4 describes Laplace's method, the Expectation Propagation approximation, and Markov chain Monte Carlo sampling for approximate inference in Gaussian process models.

The most common and successful application of Gaussian processes models is in regression problems where the noise is assumed to be homoscedastic and distributed according to a normal distribution. In practical data analysis this assumption is often inappropriate and inference is sensitive to the occurrence of more extreme errors (so called outliers). Chapter 5 proposes several variants of Gaussian process models for robust regression and describes how Bayesian inference can be approximated in each. Experiments on several data sets are presented in which the proposed models are compared with respect to their predictive performance and practical applicability. Parts of the presentation are based on Kuss et al. (2005b).

Gaussian process priors can also be used to define flexible, probabilistic classification models. Again, exact Bayesian inference is analytically intractable and various approximation techniques have been proposed, but no clear picture has yet emerged, as to when and why which algorithm should be preferred. Chapter 6 presents a detailed examination of the model, focusing on the question which approximation technique is most appropriate by investigating the structure of the posterior distribution. An experimental study is presented which corroborates the theoretical insights. This chapter is based on Kuss and Rasmussen (2005, 2006).

Reinforcement learning deals with the problem of how an agent can optimise its behaviour in a sequential decision process such that its utility over time is maximised. Chapter 7 addresses applications of Gaussian processes for model-based reinforcement learning in continuous domains. If the environment's response to the agent's actions can be predicted using Gaussian process regression models, probabilistic planning and an approximate policy iteration algorithm can be implemented. A core concept in reinforcement learning is the value function, which describes the long-term strategic value of a state. Using Gaussian process models we are able to solve an approximate continuous equivalent of the Bellman equations, and it will be shown how this can be used to estimate value functions. Parts of this chapter are based on the ideas contained in Rasmussen and Kuss (2004).

2. Bayesian Analysis

The Bayesian approach to inference constitutes the framework in which Gaussian process models will be presented in the following chapters. Even though meeting the requirements of the Bayesian approach is often difficult in practice, its principles provide a coherent guidance.

The most elementary concept in any probabilistic statistical inference is the probability of an event, e.g. the correctness of a hypothesis. Bayesian inference is founded on the theory of subjective probability, by which the probability of an event equals the subjective degree of belief in its occurrence. Cox (1946) proposed a set of postulates for a calculus of reasonable beliefs:[1]

- The degree of belief in a statement can be represented by a real number, whereby the scaling between the extremes of certainty and impossibility can be chosen arbitrarily.
- The degree of belief in $C \wedge B|A$ is in some way determined by the beliefs in $B|A$ and $C|B \wedge A$.
- The plausibility of $\neg B|A$ is determined in some way by the belief in $B|A$.

From these, Cox derived the rules of mathematical probability theory which are formally identical to the rules derived from Kolmogorov's system of probability. Furthermore, what is known as Cox's theorem states that any calculus of reasonable beliefs that meets the above postulates is equivalent to the subjective probability model. Jaynes (2003) presents a detailed elaboration suggesting the interpretation of subjective probability theory as extended logic.

A competing approach is the frequency interpretation of probability by which the probability of an event equals the relative frequency in which an event occurs in a repeated experiment in the limit of large numbers of repetitions, see for example Fisher (1922) or the discussion by Kendall (1952, ch. 7). This interpretation has been called the frequentist, orthodox, Fisherian or piscatorial probability.

Both views have their respective conceptual advantages and practical shortcomings. Subjective probability requires a person to state completely consistent beliefs about arbitrarily complex systems of hypotheses. But it allows stating the probability of an event that eludes repetition, e.g. the probability that it will rain at a given day. Frequentists object that the use of subjective probability leads to arbitrariness and incomparableness of scientific conclusions. It should be stressed that both approaches

[1] Let A, B, and C denote propositions, and let $\neg A$, $A \wedge B$, and $A|B$ be given by the common notation of Boolean algebra.

reach the same conclusions asymptotically and apart from the respective interpretation, the mathematical laws of probability calculus apply to both notions of probability.

It is important to realise that randomness is not a physical property, e.g. like mass or temperature, but probabilities only describe beliefs of the beholder. Jaynes (2003, ch. 1.8.1) cautions against the "mind projection fallacy", by which "one's own private thoughts and sensations" are interpreted as "realities existing externally in Nature".

The most apparent difference between the Bayesian and the frequentist tradition is the attribution of randomness in a statistical model. From the Bayesian point of view probabilities are used to represent uncertainties about quantities which are not directly observable, e.g. parameters of a model. However, adherents of the frequentist viewpoint argue that only the data is (partially) random and that the values of the parameters are fixed and therefore they should not be considered random variables.

2.1. Bayesian Inference

The starting point of any Bayesian analysis is a *probabilistic model* $\mathcal{M}$ which is a hypothesis about the generative process giving rise to observable data $\mathcal{D}$. A probabilistic model is a construction that describes dependencies between its elements in terms of probabilities. Within this construction there are parameters $\boldsymbol{\phi}$ which are the object of interest. Since the true value of these parameters is uncertain, $\boldsymbol{\phi}$ is a random variable. In the following it will be assumed that the parameters are continuous.

Before $\mathcal{D}$ is observed the *sampling distribution* $p(\mathcal{D}|\boldsymbol{\phi}, \mathcal{M})$ describes the predictive distribution of observations for a particular value of $\boldsymbol{\phi}$. Thereby the model is *generative*, i.e. it can be used to generate synthetic data. As a function of $\boldsymbol{\phi}$ for observed $\mathcal{D}$ we refer to $p(\mathcal{D}|\boldsymbol{\phi}, \mathcal{M})$ as the *likelihood* of the parameters. The likelihood equals the probability (density) function of the sampling distribution evaluated on the observed data. A synonymous term is direct probability.

The *prior* distribution $p(\boldsymbol{\phi}|\mathcal{M})$ describes beliefs and uncertainties about the true values of the model parameters previous to an inference step. By inference we refer to the process of combining the information contained in $\mathcal{D}$ and the prior $p(\boldsymbol{\phi}|\mathcal{M})$ into a *posterior* distribution $p(\boldsymbol{\phi}|\mathcal{D}, \mathcal{M})$. The posterior is obtained according to Bayes' rule

$$p(\boldsymbol{\phi}|\mathcal{D}, \mathcal{M}) = \frac{p(\mathcal{D}|\boldsymbol{\phi}, \mathcal{M})\, p(\boldsymbol{\phi}|\mathcal{M})}{p(\mathcal{D}|\mathcal{M})} \qquad (2.1)$$

which describes how the information contained in observations $\mathcal{D}$ changes the beliefs about $\boldsymbol{\phi}$. Intuitively, Bayes' rule can be interpreted as a weighting, in which prior beliefs about $\boldsymbol{\phi}$ are weighted according to their compatibility with the observed data. So the prior and the posterior are probability distributions describing two states relative to an inference step and correspond to potentially different beliefs about the value of $\boldsymbol{\phi}$ in $\mathcal{M}$ which gave rise to the observed data. Note that formally Bayes' rule follows directly from the definition of conditional probability.

The normalising constant appearing in the denominator of Bayes' rule

$$p(\mathcal{D}|\mathcal{M}) \;=\; \int p(\mathcal{D}|\boldsymbol{\phi}, \mathcal{M})\, p(\boldsymbol{\phi}|\mathcal{M})\, d\boldsymbol{\phi} \tag{2.2}$$

is referred to as the *evidence* or *marginal likelihood.* This quantity is of central importance for Bayesian model comparison as will be described in Section 2.3.

Any real application of Bayesian methods must acknowledge that both the prior and the likelihood have only been specified as more or less convenient approximation to the beliefs of the analyst. The prior distribution is often seen as the most controversial element of Bayesian statistics. In principle the prior should be found by introspection and consideration of all available factual information about $\boldsymbol{\phi}$ before taking the data $\mathcal{D}$ into account. However, this constitutes a non-trivial task and several approaches have been suggested to help formulate priors in practice. The prior is usually chosen from a parametric family of distributions or mixtures of those. This often gives a reasonable compromise between an accurate representation of prior beliefs and analytical tractability. In practice it is usually helpful to generate samples from the prior and inspect whether the values are reasonable *a priori.* Furthermore, the corresponding sampling distributions can be used to generate synthetic data sets whose properties should also conform with prior beliefs. Nevertheless, in any Bayesian analysis it is recommendable to examine how sensitive the posterior reacts to changes of the prior.

The more data is available and the more informative the data is about the parameters $\boldsymbol{\phi}$, the less influential the prior will be. Comparing posteriors and priors can illustrate how informative the observed data is about the parameters. When the data does not reduce the uncertainty about a parameter then both distributions will be the same, expressing that the beliefs are unchanged. Care should be taken to put non-zero prior probability on all conceivable parameter values unless one knows for certain that they are impossible (Cromwell's dictum, O'Hagan (1994, ch. 3.19)).

For models with more than just a few parameters it can become difficult to assess that a prior distribution represents precisely the prior beliefs and available factual information. If the prior information can be formulated as constraints on the parameters, a prior can be found by maximising the entropy over a suitable class of distributions subject to the parameter constraints (Jaynes, 2003, ch. 12). A prior is called *improper* if it cannot be normalised properly, e.g. a uniform distribution of infinite width. This—usually avoidable and unnecessary case—must be handled carefully, and it must be verified that the posterior is a proper distribution. A parametric family of prior distributions is called *conjugate* to a likelihood model if the resulting posterior is of the same family of distributions as the prior, regardless of the observed data.

As it is difficult to express beliefs in terms of a distribution, it is often also difficult to formulate a prior which expresses maximal uncertainty or indifference about $\boldsymbol{\phi}$, i.e. a non-informative prior. For instance, for a location parameter an improper uniform prior over an infinite range of values represents maximal uncertainty, but note that in general a uniform prior is not uninformative. Simple re-parameterisation of the likelihood changes the beliefs expressed by the prior, if the prior stays uniform. Jeffreys proposed a likelihood dependent prior which takes the re-parameterisation argument into account, the so called Jeffreys' prior (Jeffreys, 1946).

In data analysis one should always be aware of the model's assumptions, and the conclusions drawn from an analysis should obviously not be trusted more than the assumptions they are based on, whether we apply Bayesian inference or any other statistical method.

In the course of a Bayesian analysis it is often necessary to make approximations at various stages. The first step usually involves the approximation of prior beliefs by choosing a particular type of likelihood and prior distribution. Subsequently the posterior often cannot be computed analytically and approximations must be used. The errors resulting from each of the approximations can accumulate and the resulting conclusions can become invalid. It is therefore of central importance to be conscious about the accuracy of the approximations and to be as precise as possible when specifying prior distributions. It should be stressed that the Bayesian framework can be sensitive to inaccurate specifications of the prior, which can become a severe problem especially in situations with complex models and few data.

2.2. Bayesian Decision Theory

A Bayesian analysis typically proceeds in two stages. At first the posterior distribution $p(\boldsymbol{\phi}|\mathcal{D},\mathcal{M})$ is computed which summarises all available information about $\boldsymbol{\phi}$. Having obtained the posterior, it can be used to deduce further statements about the true value of $\boldsymbol{\phi}$. The degree of belief in any statement regarding the parameters can be computed by evaluating the probability of the statement under the posterior distribution.

The most common problem is to state a single point estimate of $\boldsymbol{\phi}$ which in the Bayesian framework is considered a decision problem. Let $\hat{\boldsymbol{\phi}}$ be a point estimate of the true value of $\boldsymbol{\phi}$. The loss function $\mathsf{L}(\hat{\boldsymbol{\phi}},\boldsymbol{\phi})$ characterises the loss associated with a discrepancy between the point estimate and the unknown true parameter value. The risk of a decision is the expected loss

$$\mathsf{R}_{\mathsf{L}}(\hat{\boldsymbol{\phi}}) \;=\; \int \mathsf{L}(\hat{\boldsymbol{\phi}},\boldsymbol{\phi})\, p(\boldsymbol{\phi}|\mathcal{D},\mathcal{M})d\boldsymbol{\phi} \tag{2.3}$$

where the expectation is taken with respect to the posterior distribution. The optimal Bayesian decision is the point estimate

$$\boldsymbol{\phi}^{\star} \;=\; \operatorname*{argmin}_{\hat{\boldsymbol{\phi}}} \mathsf{R}_{\mathsf{L}}(\hat{\boldsymbol{\phi}}) \;, \tag{2.4}$$

which minimises the risk. For example, the expected absolute error $\mathsf{L}(\hat{\boldsymbol{\phi}},\boldsymbol{\phi}) = |\hat{\boldsymbol{\phi}}-\boldsymbol{\phi}|$ is minimised by the median of the posterior distribution. Likewise minimising the expected squared error $\mathsf{L}(\hat{\boldsymbol{\phi}},\boldsymbol{\phi}) = (\hat{\boldsymbol{\phi}}-\boldsymbol{\phi})^2$ leads to choosing the mean of the posterior.

The mode of the posterior, which will be referred to as the *maximum a posteriori* (MAP) estimate, is optimal given a loss function which is zero if the estimate and the true value match exactly and a positive constant otherwise (Jaynes, 2003, ch. 13.9).

Often the MAP estimate is relatively easy to compute

$$\boldsymbol{\phi}^\star_{\mathrm{MAP}} \;=\; \underset{\boldsymbol{\phi}}{\operatorname{argmax}}\, p(\boldsymbol{\phi}|\mathcal{D}) \;=\; \underset{\boldsymbol{\phi}}{\operatorname{argmax}}\, p(\mathcal{D}|\boldsymbol{\phi})p(\boldsymbol{\phi}) \tag{2.5}$$

since it can be found without normalising the posterior. The MAP estimate is also known as the penalised maximum likelihood estimate. Quite contrary to the outstanding rôle of the maximum likelihood estimate in frequentist statistics, the MAP estimate is subject to severe criticism from a Bayesian perspective. The MAP focuses on the posterior *density* and neglects posterior *mass* and can therefore be very atypical for the posterior distribution. Equivalently, the mode of the posterior might be atypical for the whole distribution (an example thereof will be encountered in Chapter 6). From a practical point of view the loss function minimised by the MAP estimate is questionable because it states that if an error is made it does not matter how large the error is. A further point of criticism is that by re-parameterisation of the model the MAP estimate can be changed arbitrarily. Assume continuous parameters $\boldsymbol{\phi}$ and let $\boldsymbol{\xi} = h(\boldsymbol{\phi})$ denote a bijective (one-to-one) function used for mapping of parameters between the parameterisations. The maximum *a posteriori* estimate of $\boldsymbol{\xi}$ is

$$\boldsymbol{\xi}^\star_{MAP} \;=\; \underset{\boldsymbol{\xi}}{\operatorname{argmax}}\, p(\boldsymbol{\xi}|\mathcal{D}) \;=\; \underset{\boldsymbol{\xi}}{\operatorname{argmax}}\, p(h^{-1}(\boldsymbol{\xi})|\mathcal{D}) \left|\frac{\partial h^{-1}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}^\top}\right| \tag{2.6}$$

where the last term on the right hand side is the Jacobi determinant of the transformation $\boldsymbol{\phi} = h^{-1}(\boldsymbol{\xi})$ (DeGroot and Schervish, 2002, ch. 3.9). By comparing estimates (2.6) and (2.5) it becomes obvious why for general bijective re-parameterisation h the MAP estimate $\boldsymbol{\xi}^\star$ will not be identical to $h(\boldsymbol{\phi}^\star)$. Note that this does not hold for maximum likelihood estimates, which are invariant to re-parameterisation, since a likelihood is not a density and therefore no Jacobian correction is necessary. This is not unique to maximum likelihood estimates, since in general Bayesian inference is invariant to re-parameterisations of the likelihood, if both the loss function and the prior distributions are correctly transformed as well.

We described above how a point estimate of parameters $\boldsymbol{\phi}$ can be found using Bayesian decision theory. This approach is not limited to decisions about model parameters but can also be used to derive statements about any random variable. In particular, we will use models to make predictive statements about unobserved quantities. Let y_* denote such an unobserved quantity and let $p(y_*|\boldsymbol{\phi}, \mathcal{M})$ describe the dependency of y_* on $\boldsymbol{\phi}$ given by $\mathcal{M}$. The posterior predictive distribution of y_* is found by averaging over the posterior uncertainty in the model's parameters

$$p(y_*|\mathcal{D}, \mathcal{M}) \;=\; \int p(y_*|\boldsymbol{\phi}, \mathcal{M})\, p(\boldsymbol{\phi}|\mathcal{D}, \mathcal{M})\, d\boldsymbol{\phi}\,. \tag{2.7}$$

The predictive distribution $p(y_*|\mathcal{D}, \mathcal{M})$ describes the posterior beliefs about y_*. If we are asked to give a point estimate $\hat{y}_*$ of y_*, we again make use of Bayesian decision theory. Given a loss function the optimal (point) prediction is found by minimising the risk, which is now a loss function averaged over the predictive distribution (2.7).

2.3. Model Comparison and Model Selection

In the previous sections, Bayesian inference was used to describe how observed data changes the beliefs and uncertainties about the values of parameters in a given model. The idea can be applied at a higher level addressing uncertainty about the model itself. The model $\mathcal{M}$ can be interpreted as a hypothesis about the data-generating process and refers to both a likelihood and a prior on the parameters.

Model comparison and hypothesis testing are conceptually identical in the Bayesian framework, since different models can be understood as different hypotheses about the data-generating process. Selecting a single best model corresponds to a decision problem, analogous to finding a point estimate of a parameter.

2.3.1. Bayesian Model Comparison

Assume a finite number $i = 1, \dots, N$ of hypotheses $\mathcal{M}_i$ about the data-generating process. Let $p(\mathcal{M}_i)$ denote the prior belief in model i. The posterior belief in $\mathcal{M}_i$ is obtained by applying Bayes' rule

$$p(\mathcal{M}_i|\mathcal{D}) \;\propto\; p(\mathcal{D}|\mathcal{M}_i)\, p(\mathcal{M}_i)\,. \tag{2.8}$$

Making a pairwise comparison of models A and B avoids computing the normalising constant $p(\mathcal{D})$ and the ratio

$$\frac{p(\mathcal{M}_A|\mathcal{D})}{p(\mathcal{M}_B|\mathcal{D})} = \frac{p(\mathcal{D}|\mathcal{M}_A)}{p(\mathcal{D}|\mathcal{M}_B)}\,\frac{p(\mathcal{M}_A)}{p(\mathcal{M}_B)} \tag{2.9}$$

is called the posterior odds. Assuming the prior $p(\mathcal{M}_i)$ to be uniform over all models, the posterior odds equal the ratio of evidences which is called Bayes factor (Good, 1958). The Bayes factor quantifies the relative plausibility of model A compared to model B.

Moreover, if the prior $p(\mathcal{M}_i)$ is uniform over all models, the posterior belief in a model (2.8) is proportional to its evidence. Thereby the evidence allows us to rank models according to the posterior belief.

The central rôle of the evidence for model comparison motivates a closer look at this quantity. The synonym *marginal likelihood* refers to the interpretation that $p(\mathcal{D}|\mathcal{M})$ is obtained by marginalisation of the joint probability $p(\mathcal{D}, \boldsymbol{\phi}|\mathcal{M})$ over $\boldsymbol{\phi}$ as in eq. (2.2). Before the data is observed $p(\mathcal{D}|\mathcal{M})$ can be interpreted as an *a priori* predictive distribution of the data. Therefore, once the data becomes available $p(\mathcal{D}|\mathcal{M})$ measures the *expectedness* of the data under a given model. Note that $p(\mathcal{D}|\mathcal{M})$ must normalise properly, so that if a model gives high predictive probability to certain data sets it must give less to others. A very flexible (versatile or complex) model can generate a wide range of data sets but the predictive probability for a particular data set will be small. A model that explains only a small set of possible observations will be considered very plausible if such a data set is actually observed. Therefore, if there are several possible explanations of a data set, the simpler model is favoured over an alternative complex explanation, i.e. the model for which the observed data appears most typical.

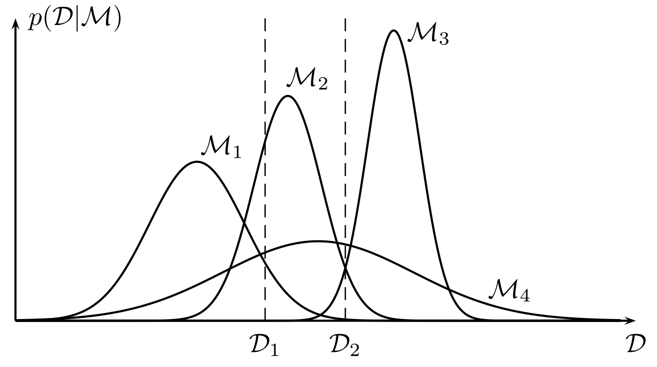


Figure 2.1.: Rôle of the evidence for model comparison. The abscissa abstractly represents all possible data sets $\mathcal{D}$ of a given size. Shown are the evidence $p(\mathcal{D}|\mathcal{M}_i)$ as a function of $\mathcal{D}$ for four models $\mathcal{M}_1, \dots, \mathcal{M}_4$. The dashed lines mark two data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ that could be observed. Model $\mathcal{M}_4$ covers a wide range of data sets representing the most flexible model. Model $\mathcal{M}_3$ is the most constricted, being able to generate a smaller variety of data sets. If the data set $\mathcal{D}_1$ had been observed, the most plausible model would be $\mathcal{M}_2$ which would be roughly twice as plausible as $\mathcal{M}_1$ and $\mathcal{M}_4$. Since $\mathcal{D}_1$ would be highly implausible under $\mathcal{M}_3$ the posterior belief in this model would be very small. If instead data $\mathcal{D}_2$ were observed, the posterior uncertainty would remain relatively large with models $\mathcal{M}_2$, $\mathcal{M}_3$, and $\mathcal{M}_4$ being almost equally plausible and only $\mathcal{M}_1$ seeming unlikely. The illustration is similar to Figure 2 in MacKay (1992b).

This characteristic of Bayesian model comparison has been interpreted as an instance of "Occam's Razor" which is often quoted as "Entities are not to be multiplied without necessity" and attributed to Willam of Ockham (approx. 1287–1347). Figure 2.1 illustrates the rôle of the evidence in Bayesian model comparison. Note that the use of the terms *flexible* and *simple* was associated with the expressiveness of a probabilistic model, i.e. the variability of data sets that a model can generate, and not the number of parameters as in other criteria for model selection, see Kass and Raftery (1995) for a discussion.

The predictive nature of the evidence can also be understood by rewriting it in a sequential way. Let $\mathcal{D} = \{x_1, \dots, x_m\}$ where the elements x are individual observations. By rewriting the evidence as a product of conditional distributions we obtain

$$p(\mathcal{D}|\mathcal{M}) \;=\; p(x_1, \dots, x_m|\mathcal{M}) \;=\; \prod_{i=1}^{m} p(x_i|\{x_j\}_{j<i}, \mathcal{M}) \tag{2.10}$$

where each term $p(x_i|\{x_j\}_{j<i}, \mathcal{M})$ can be interpreted as the posterior predictive distribution of x_i given the observations $x_1, \dots, x_{i-1}$ and the model $\mathcal{M}$. If the x are exchangeable the argument is invariant to the order of observations. So if observations can be well predicted from "previous" observations, the evidence for the model increases.

In the machine learning context models are usually used for prediction. Assume N

models, all able to produce a predictive distribution $p(y_*|\mathcal{M}_i, \mathcal{D})$, as for example given by eq. (2.7). The predictions of different models are then weighted by the posterior belief in the respective model

$$p(y_*|\mathcal{D}) \;=\; \sum_{i=1}^{N} p(y_*|\mathcal{M}_i, \mathcal{D})\, p(\mathcal{M}_i|\mathcal{D}) \;, \tag{2.11}$$

which is an averaging over the posterior model uncertainty.

In judging the appropriateness of a model two desiderata have to be balanced: the ability to explain the observed data and the ability to generalise to the whole population. For the latter, prior beliefs and the choice of model become decisive. A common problem when seeking a single explanation of the observed data, e.g. a maximum likelihood estimate in frequentist statistics or a neural network trained by backpropagation, is the bias-variance-dilemma and its effects are known as *under-fitting* and *over-fitting*. Assume that the data-generating process is decomposable into a systematic (deterministic) and a random component, as for example assumed in regression analysis. If a model of the systematic component is flexible enough, a maximum likelihood estimate can misinterpret random fluctuation as systematic variation, i.e. over-fitting may occur. Likewise, if a model makes too simplistic assumptions, systematic variation can be interpreted as random fluctuation which is known as under-fitting. In the Bayesian framework these problems are avoided—in principle—by not picking a single explanation but by weighted averaging according to the posterior beliefs on both the parameter (2.7) and model level (2.11).

2.3.2. Model Selection by Evidence Maximisation

The previous section described how model uncertainty can be handled conceptually. In situations where handling multiple models is computationally prohibitive, a common approach is to select a single model which appears to be most plausible given the observed data. Assuming the prior $p(\mathcal{M}_i)$ to be uniform over all models this corresponds to selecting the model showing the highest evidence.

In the above description a model $\mathcal{M}$ represented both a sampling distribution, i.e. a likelihood and a fixed prior on its parameters. If the likelihood is considered to be correct with certainty, then from a dogmatic point of view no model comparison is necessary since the prior should reflect beliefs independent of $\mathcal{D}$. However, practically the same formalism can be used to implement what could be called *prior selection*.

Suppose the priors under consideration all belong to a parametric family $p(\boldsymbol{\phi}|\boldsymbol{\psi})$ where $\boldsymbol{\psi}$ are referred to as hyper-parameters. In principle, the use of hyper-parameters is an unnecessary construction since given a hyper-prior $p(\boldsymbol{\psi})$ they can always be integrated out

$$p(\boldsymbol{\phi}) \;=\; \int p(\boldsymbol{\phi}|\boldsymbol{\psi})\, p(\boldsymbol{\psi})\, d\boldsymbol{\psi}\,. \tag{2.12}$$

However, in many situations it has shown to be easier to construct a hierarchical structure to represent prior beliefs, because it allows a more intuitive description of dependencies between parameters, see for example Gelman et al. (1995, ch. 5). The problem of prior selection can be seen as finding the value of $\boldsymbol{\psi}$ that maximises the evidence

$$p(\mathcal{D}|\boldsymbol{\psi}) \;=\; \int p(\mathcal{D}|\boldsymbol{\phi})\, p(\boldsymbol{\phi}|\boldsymbol{\psi})\, d\boldsymbol{\phi} \tag{2.13}$$

which is now conditioned on the hyper-parameters. Ignoring conceptual objections eq. (2.13) has the form of a likelihood for $\boldsymbol{\psi}$ and

$$\boldsymbol{\psi}^\star \;=\; \underset{\boldsymbol{\psi}}{\operatorname{argmax}}\; p(\mathcal{D}|\boldsymbol{\psi}) \tag{2.14}$$

gives the *maximum likelihood type II* (ML-II) estimate of $\boldsymbol{\psi}$. The corresponding $p(\boldsymbol{\phi}|\boldsymbol{\psi}^\star)$ is called the maximum likelihood type II prior and $\boldsymbol{\psi}^\star$ are referred to as the ML-II hyper-parameters. Intuitively, the ML-II prior is the prior that would have made the observed data most expected given the assumed sampling distribution, i.e. the likelihood. This approach is also known as *empirical Bayes*, although this name is somewhat misleading because Bayesian inference itself is always empirical, see Gelman et al. (1995, ch. 5), O'Hagan (1994, chs. 5.25ff), or Berger (1985, chs. 3.5.4 & 4.5) for details.

From a Bayesian point of view, selecting a single model is a decision problem as described in Section 2.2. The ML-II estimate can be interpreted as the MAP estimate of

$$p(\boldsymbol{\psi}|\mathcal{D}) \;\propto\; p(\mathcal{D}|\boldsymbol{\psi})\, p(\boldsymbol{\psi}) \tag{2.15}$$

when $p(\boldsymbol{\psi})$ is uniform, i.e. a constant and degenerate prior. Note that the MAP estimate depends on the parameterisation as described in Section 2.2.

In principle, Bayesian inference can proceed almost automatically once a likelihood and a prior are specified. But in practice the analysis is often complicated by the necessity to solve integrals which are analytically intractable. In the following chapters a common situation will be that Bayesian inference can be performed analytically for a subset of parameters and the remaining parameters and hyper-parameters will be set to their ML-II estimated values. Let the parameters be divided into $\boldsymbol{\phi} = [\boldsymbol{\phi}_A, \boldsymbol{\phi}_B]$ such that the conditional posterior of $\boldsymbol{\phi}_B$ can be computed

$$p(\boldsymbol{\phi}_B|\boldsymbol{\phi}_A, \boldsymbol{\psi}, \mathcal{D}) \;=\; \frac{p(\mathcal{D}|\boldsymbol{\phi}_B, \boldsymbol{\phi}_A)\, p(\boldsymbol{\phi}_B|\boldsymbol{\psi})}{p(\mathcal{D}|\boldsymbol{\phi}_A, \boldsymbol{\psi})}\,, \tag{2.16}$$

but inference over $\boldsymbol{\phi}_A$ and $\boldsymbol{\psi}$ is intractable. Nevertheless, maximum likelihood type II estimates can be computed

$$(\boldsymbol{\phi}_A^\star, \boldsymbol{\psi}^\star) \;=\; \underset{\boldsymbol{\phi}_A, \boldsymbol{\psi}}{\operatorname{argmax}}\; p(\mathcal{D}|\boldsymbol{\phi}_A, \boldsymbol{\psi}) \tag{2.17}$$

by maximising the evidence, i.e. the denominator in eq. (2.16):

$$p(\mathcal{D}|\boldsymbol{\phi}_A, \boldsymbol{\psi}) \;=\; \int p(\mathcal{D}|\boldsymbol{\phi}_A, \boldsymbol{\phi}_B)\, p(\boldsymbol{\phi}_B|\boldsymbol{\psi})\, d\boldsymbol{\phi}_B\,. \tag{2.18}$$

If for example $\boldsymbol{\phi}_A$ and $\boldsymbol{\psi}$ are nuisance parameters, the marginal posterior of $\boldsymbol{\phi}_B$ can be approximated by conditioning on the ML-II estimates of the parameters:

$$p(\boldsymbol{\phi}_B|\mathcal{D}) \;=\; \int p(\boldsymbol{\phi}_A, \boldsymbol{\phi}_B, \boldsymbol{\psi}|\mathcal{D})\, d\boldsymbol{\phi}_A\, d\boldsymbol{\psi} \;\approx\; p(\boldsymbol{\phi}_B|\boldsymbol{\phi}_A^\star, \boldsymbol{\psi}^\star, \mathcal{D})\,. \tag{2.19}$$

The evidence has been praised for the built-in compromise between model complexity and ability to explain the data. However, optimising the prior or likelihood parameters in the light of the observed data can lead to serious over-fitting, as we will observe in later chapters.

2.4. Bibliographical Remarks

The Bayesian approach is named after Thomas Bayes (1702–1761) whose “Essay towards solving a problem in the doctrine of chances” was posthumously published in 1763. The attribution is somewhat misleading since the so called Bayes’ rule (2.1) cannot be found explicitly in his work. As Jaynes (2003, ch. 4.6.1) points out, Bayes’ rule follows directly from the definition of conditional probability which had been described long before, e.g. by Bernoulli and de Moivre. But it was Laplace (1774) who described its use for inference in general terms in his “Mémoire sur la probabilité des causes par les évènemens”. Before this approach has been named Bayesian it was known as the method of inverse probability, describing the posterior, in contrast to direct probability, the likelihood (Kendall, 1952, ch. 7).

The development of the statistical science gained accelerated momentum in the early 20th century. In this phase the frequentist approach was almost uncontested in the statistical science. Since around the 1960’s a revival of Bayesian ideas was stimulated amongst others by physicists like Harold Jeffreys. Many books about the historical development of probability theory and statistics exist of which Stigler (1999) is a recommendable example. Jaynes (1986) and Fienberg (2006) give historical reviews of Bayesian analysis and the development of its core concepts. Bayes’ original 1763 essay is reprinted with an introduction by Barnard (1958). A summary of Laplace’s 1774 “Mémoire sur la probabilité des causes par les évènemens” can be found in Stigler (1986). The relation of Occam’s razor and Bayesian reasoning is examined in more detail by Jeffreys and Berger (1992) and Rasmussen and Ghahramani (2001). Good (1958) introduced the Bayes factor which is further described by Kass and Raftery (1995). Maximum likelihood II estimation is discussed in detail by MacKay (1999a) and Berger (1985, ch. 4.5). A comparison of Bayesian and frequentist approaches to parameter estimation and model selection is given by Ripley (1998).

General references regarding Bayesian methods include Box and Tiao (1973), O’Hagan (1994), Bernardo and Smith (1994), Gelman et al. (1995), and Jaynes (2003). Berger (1985) and Robert (1994) approach Bayesian inference from a decision theoretic point of view.

3. Gaussian Process Models

> If I were actively concerned with the analysis of data from stochastic processes, I believe that I should try to seek out techniques of data processing which were not too closely tied to individual models, which might be likely to be unexpectedly revealing, and which were being pushed by the needs of actual data analysis.
>
> — Tukey (1962)

This chapter introduces the class of Gaussian process models of which several instances will be studied in the following chapters. Common to all these models is that they assume a directional dependency between an input or covariate $\mathbf{x}$ and the corresponding observable output or response y. Based on empirical observations the model attempts to describe the conditional distribution $p(y|\mathbf{x})$. If $y \in \mathbb{R}$, then $p(y|\mathbf{x})$ is called a regression model and if y is nominal, then it is called a classification model.

3.1. Structure of Gaussian Process Models

The conditional distribution $p(y|\mathbf{x})$ describes the dependency of an observable y on a corresponding input $\mathbf{x} \in \mathcal{X}$. The class of models described in this section assumes that this relation can be decomposed into a systematic and a random component. Furthermore, the systematic dependency is given by a latent function $f : \mathcal{X} \to \mathbb{R}$ such that the sampling distribution, i.e. the likelihood, is of the form

$$p(y|f(\mathbf{x}), \boldsymbol{\theta}) \;, \tag{3.1}$$

which describes the random aspects of the data-generating process. In general, we will use $\boldsymbol{\theta}$ to denote additional parameters of the likelihood besides f. Note that the conditional distribution of the observable y depends on $\mathbf{x}$ via the value of the latent function f at $\mathbf{x}$ only.

An example is a regression model where the systematic relation $f(\mathbf{x})$ cannot be observed directly but only noisy samples thereof. For instance the sampling distribution $p(y|f(\mathbf{x}), \boldsymbol{\theta}) = \mathcal{N}(y|f(\mathbf{x}), \sigma_{\mathtt{n}}^2)$ describes regression with normal noise where the noise variance $\sigma_{\mathtt{n}}^2$ is an additional parameter, such that $\boldsymbol{\theta} = \sigma_{\mathtt{n}}^2$. However, models of the form of eq. (3.1) are not limited to regression models, as will be shown in Chapter 6.

The aim of inference is to identify the systematic component f from empirical observations and prior beliefs. The data comes in the form of pairwise observations $\mathcal{D} = \{(y_i, \mathbf{x}_i)|i = 1, \ldots, m\}$ where m is the number of samples. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^\top$ and $\mathbf{y} = [y_1, \ldots, y_m]^\top$ collect the inputs and responses respectively. In general we assume $\mathbf{x} \in \mathbb{R}^n$ unless stated otherwise. In the following we will distinguish between *training* data which is used for model selection and inference and *test* data for which the model is used to make predictions, e.g. in order to assess the model's generalisation ability.

In order to make inference about f we need to formalise a prior belief about the latent function. The parametric approach is to assume a structure for $f(\mathbf{x}, \boldsymbol{\phi})$ with finitely many parameters $\boldsymbol{\phi}$, e.g. an artificial neural network or a classical linear model $f(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\phi}$. In this case the prior uncertainty about f is usually expressed in terms of a prior distribution on $\boldsymbol{\phi}$. However, in situations where the form of f is not known, assuming a particular parametric form might be too restrictive. The drawback of parametric models is that the accuracy by which f can be identified is bounded by the best function having the assumed structure. However, in the end we are interested in inference about the function f and not about parameters. Therefore it is an intuitively appealing approach to make inference about f directly, i.e. in a non-parametric way.

The name *Gaussian process model* refers to using a Gaussian process (GP) as a prior on f. The Gaussian process prior is non-parametric in the sense that instead of assuming a particular parametric form of $f(\mathbf{x}, \boldsymbol{\phi})$ and making inference about $\boldsymbol{\phi}$, the approach is to put a prior on function values directly.[1] Each input position $\mathbf{x} \in \mathcal{X}$ has an associated random variable $f(\mathbf{x})$. A Gaussian process prior on f technically means that *a priori* the joint distribution of a collection of function values $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{m'})]^\top$ associated with any collection of m' inputs $[\mathbf{x}_1, \ldots, \mathbf{x}_{m'}]^\top$ is multivariate normal (Gaussian)

$$p(\mathbf{f}|\mathbf{X}, \boldsymbol{\psi}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{K}) \tag{3.2}$$

with mean $\mathbf{m}$ and covariance matrix $\mathbf{K}$. A Gaussian process is specified by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}', \boldsymbol{\psi})$ such that $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{\psi})$ and $\mathbf{m} = [m(\mathbf{x}_1), \ldots, m(\mathbf{x}_{m'})]^\top$.[2] By choosing a particular form of covariance function we may introduce hyper-parameters $\boldsymbol{\psi}$ to the Gaussian process prior. Depending on the actual form of the covariance function $k(\mathbf{x}, \mathbf{x}', \boldsymbol{\psi})$ the hyper-parameters $\boldsymbol{\psi}$ can control various aspects of the Gaussian process, as will be described in Section 3.3. For notational simplicity, below we will often write the covariance function $k(\mathbf{x}, \mathbf{x}')$ without writing the dependency of k on further parameters $\boldsymbol{\psi}$ explicitly.

Note that the (sampling) distribution (3.1) of y depends on f only through $f(\mathbf{x})$. As

[1]The term *non-parametric*, as it is used here, is only meant to emphasise that a prior is put directly on functions without explicitly parameterising the unknown function. Using a prior directly on the space of functions corresponds to an infinite dimensional parameter space. Making inference about an infinite number of parameters seems impractical, but it turns out that for any given data set only a finite number of parameters (function values) has to be represented explicitly. However, this *effective* number of parameters depends on the number of observations, which is a typical characteristic of non-parametric models. Note that this is not the case for certain Gaussian process priors, depending on the covariance function, because there exists an equivalent parametric model with a fixed number of parameters, as will be described in Section 3.4.

[2]Note that throughout we use $m(\mathbf{x})$ to denote mean functions of Gaussian processes while m (without $\mathbf{x}$) refers to the number of observations.

an effect the likelihood of f given $\mathcal{D}$ factorises

$$p(\mathbf{y}|f,\boldsymbol{\theta}) \;=\; \prod_{i=1}^{m} p(y_i|f(\mathbf{x}_i),\boldsymbol{\theta}) \;=\; p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta}) \tag{3.3}$$

and depends on f only through its value at the observed inputs $\mathbf{f}$. According to the model, conditioning the likelihood on $\mathbf{f}$ is equivalent to conditioning on the full function f. This is of central importance since it allows us to make inference over finite dimensional quantities instead of handling the whole function f. The posterior distribution of the function values $\mathbf{f}$ is computed according to Bayes' rule

$$p(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) \;=\; \frac{p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta})\,p(\mathbf{f}|\mathbf{X},\boldsymbol{\psi})}{p(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})} \;=\; \frac{\mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{K})}{p(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})}\prod_{i=1}^{m} p(y_i|f_i,\boldsymbol{\theta}) \tag{3.4}$$

where $f_i = f(\mathbf{x}_i)$. If the likelihood (3.1) is log-concave in $\mathbf{f}$, the posterior is a concave, unimodal distribution (Boyd and Vandenberghe, 2004, ch. 3.5).

The posterior distribution of $\mathbf{f}$ can be used to compute the posterior predictive distribution of $f(\mathbf{x}_*)$ for any input $\mathbf{x}_*$ where in the following the asterisk is used to mark test examples. If several test cases are given, $\mathbf{X}_*$ collects the test inputs and $\mathbf{f}_*$ denotes the corresponding vector of latent function values. The predictive distribution of $\mathbf{f}_*$ is obtained by integration over the posterior uncertainty

$$p(\mathbf{f}_*|\mathcal{D},\mathbf{X}_*,\boldsymbol{\theta},\boldsymbol{\psi}) \;=\; \int p(\mathbf{f}_*|\mathbf{f},\mathbf{X},\mathbf{X}_*,\boldsymbol{\psi})\,p(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi})\,d\mathbf{f} \tag{3.5}$$

where the first term of the right hand side describes the dependency of $\mathbf{f}_*$ on $\mathbf{f}$ induced by the GP prior. The joint prior distribution of $\mathbf{f}$ and $\mathbf{f}_*$ due to the GP prior is multivariate normal

$$p(\mathbf{f},\mathbf{f}_*|\mathbf{X},\mathbf{X}_*,\boldsymbol{\psi}) \;=\; \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_*\end{bmatrix}\middle|\begin{bmatrix}\mathbf{m}\\ \mathbf{m}_*\end{bmatrix},\begin{bmatrix}\mathbf{K} & \mathbf{K}_*\\ \mathbf{K}_*^\top & \mathbf{K}_{**}\end{bmatrix}\right) \tag{3.6}$$

where the covariance matrix is partitioned such that $\mathbf{K}_{**}$ is the prior covariance matrix of the $\mathbf{f}_*$ and $\mathbf{K}_*$ contains the covariances between $\mathbf{f}$ and $\mathbf{f}_*$. The conditional distribution of $\mathbf{f}_*|\mathbf{f}$ can be obtained from the joint distribution (3.6) using relation (B.22) to give

$$p(\mathbf{f}_*|\mathbf{f},\mathbf{X},\mathbf{X}_*,\boldsymbol{\psi}) \;=\; \mathcal{N}\left(\mathbf{f}_*|\mathbf{m}_* + \mathbf{K}_*^\top\mathbf{K}^{-1}(\mathbf{f}-\mathbf{m}), \mathbf{K}_{**} - \mathbf{K}_*^\top\mathbf{K}^{-1}\mathbf{K}_*\right) \tag{3.7}$$

which is again multivariate normal.

The simplest possible model assumes that the function can be observed directly $y = f(\mathbf{x})$ so that the posterior on $\mathbf{f}$ becomes $p(\mathbf{f}|\mathcal{D}) = \delta(\mathbf{f}-\mathbf{y})$, describing that no posterior uncertainty about $\mathbf{f}$ remains. From this posterior the predictive distribution of $\mathbf{f}_*$ can be obtained according to eq. (3.5) which corresponds to simply replacing $\mathbf{f}$ by $\mathbf{y}$ in eq. (3.7). See Figure 3.1 for an illustration.

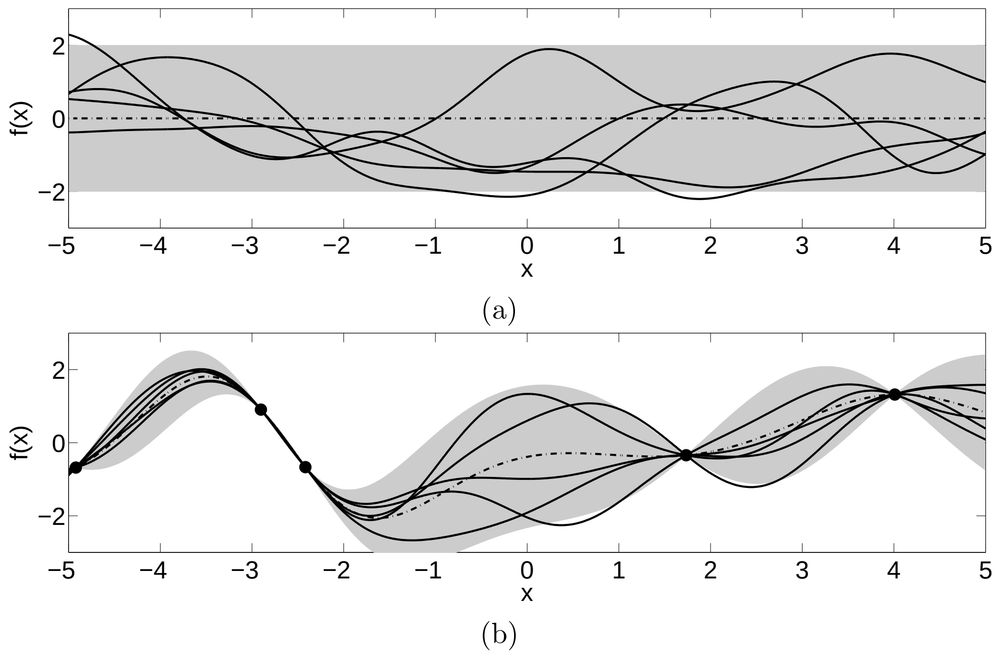


Figure 3.1.: Gaussian process regression without observational noise. Panel (a) shows a GP prior with zero mean $m(x) = 0$ and covariance function $k(x, x') = \exp(-\|x - x'\|^2 / 2)$. The grey area depicts ± 2 standard deviations of $f(x)$ and the dashed line describes the mean function. The five solid lines are sample paths from the Gaussian process (how these are generated will be described in Section 3.3.1). Panel (b) shows the posterior Gaussian process which is obtained by conditioning the prior on the five observations depicted as points. The predictive uncertainty is zero at the locations where the function value has been observed. Between the observations the uncertainty about the function value grows and the sampled functions represent valid hypothesis about f under the posterior process.

3.2. Regression with Normal Noise

Bayesian inference about the latent function f in Gaussian process regression models is analytically tractable if the observational noise is assumed to be normally distributed.[3] In this case the resulting posterior process $f|\mathcal{D}$ is again a Gaussian process so this can be considered as the conjugate setting.

The model assumes $y = f(\mathbf{x}) + \varepsilon$ where the homoscedastic additive observational error follows a normal distribution $\mathcal{N}(\varepsilon|0, \sigma_\mathfrak{n}^2)$ such that $p(y|f(\mathbf{x}), \boldsymbol{\theta}) = \mathcal{N}(y|f(\mathbf{x}), \sigma_\mathfrak{n}^2)$. The noise variance $\sigma_\mathfrak{n}^2$ is an additional parameter of the likelihood, such that $\boldsymbol{\theta} = \sigma_\mathfrak{n}^2$ below.

[3]Note that the regression model without noise described in the previous section can be seen as an instance of regression with normal Gaussian noise in the limit of zero noise variance.

Given the observed input locations $\mathbf{X}$ the likelihood of f becomes

$$p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta}) \;=\; \prod_{i=1}^{m} \mathcal{N}(y_i|f_i,\sigma_{\mathfrak{n}}^2) \;=\; \mathcal{N}(\mathbf{y}|\mathbf{f},\sigma_{\mathfrak{n}}^2\mathbf{I})\,. \tag{3.8}$$

Again a Gaussian process is used as a prior on f. For simplicity of exposition the prior mean function is set to be zero $m(\mathbf{x}) = 0$. This might agree with actual prior beliefs only if the problem is transformed accordingly, e.g. by subtracting the mean of $\mathbf{y}$ or some other form of transformation as will be described in Section 3.2.2. Note that also nonzero mean functions can be handled, e.g. deterministic functions or linear models with uncertain parameters. Assuming the mean to be zero, the prior on the latent function values at the observed inputs is

$$p(\mathbf{f}|\mathbf{X},\boldsymbol{\psi}) \;=\; \mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K}) \tag{3.9}$$

where the elements of the covariance matrix $\mathbf{K}$ are computed element-wise using a covariance function $k(\mathbf{x},\mathbf{x}',\boldsymbol{\psi})$. Since likelihood and prior are both multivariate normal, the posterior on $\mathbf{f}$ can be calculated analytically

$$\begin{aligned}
p(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) &\propto p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta})\,p(\mathbf{f}|\mathbf{X},\boldsymbol{\psi}) && (3.10a)\\
&= \mathcal{N}(\mathbf{y}|\mathbf{f},\sigma_{\mathfrak{n}}^2\mathbf{I})\,\mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K}) && (3.10b)\\
&\propto \mathcal{N}(\mathbf{f}|\mathbf{K}(\mathbf{K}+\sigma_{\mathfrak{n}}^2\mathbf{I})^{-1}\mathbf{y},(\mathbf{K}^{-1}+\sigma_{\mathfrak{n}}^{-2}\mathbf{I})^{-1}) && (3.10c)
\end{aligned}$$

and is also a multivariate normal distribution according to eq. (B.16) given in Appendix B.2.2.1.

The posterior predictive distribution of the latent function values $\mathbf{f}_*$ for an arbitrary set of test locations $\mathbf{X}_*$ can be computed according to eq. (3.5) by averaging over the posterior (3.10), where the first term of the right hand side of eq. (3.5) is given by eq. (3.7). Using eq. (B.19) the resulting predictive distribution is again multivariate normal

$$p(\mathbf{f}_*|\mathcal{D},\mathbf{X}_*,\boldsymbol{\theta},\boldsymbol{\psi}) = \mathcal{N}\left(\mathbf{f}_*|\mathbf{K}_*^\top(\mathbf{K}+\sigma_{\mathfrak{n}}^2\mathbf{I})^{-1}\mathbf{y},\mathbf{K}_{**}-\mathbf{K}_*^\top(\mathbf{K}+\sigma_{\mathfrak{n}}^2\mathbf{I})^{-1}\mathbf{K}_*\right)\,. \tag{3.11}$$

Note that the predictive uncertainty, i.e. the covariance matrix of $\mathbf{f}_*$, does not depend on $\mathbf{y}$, but only on the dependencies induced by the covariance as a function of $\mathbf{X}_*$ and $\mathbf{X}$. The predictive distribution of noisy observations $\mathbf{y}_*$ can be derived from eq. (3.11) by adding $\sigma_{\mathfrak{n}}^2\mathbf{I}$ to the posterior covariance matrix of $\mathbf{f}_*$.

The above argumentation generalises to an arbitrary set of input locations, meaning that the posterior process $f|\mathcal{D}$ is again a Gaussian process with posterior mean and covariance function

$$\begin{aligned}
m_*(\mathbf{x}) &= \mathbf{k}(\mathbf{x})^\top(\mathbf{K}+\sigma_{\mathfrak{n}}^2\mathbf{I})^{-1}\mathbf{y} && (3.12a)\\
k_*(\mathbf{x},\mathbf{x}') &= k(\mathbf{x},\mathbf{x}')-\mathbf{k}(\mathbf{x})^\top(\mathbf{K}+\sigma_{\mathfrak{n}}^2\mathbf{I})^{-1}\,\mathbf{k}(\mathbf{x}') && (3.12b)
\end{aligned}$$

where $\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}), \ldots, k(\mathbf{x}_m, \mathbf{x})]^\top$ is a vector of prior covariances between $\mathbf{x}$ and the training inputs $\mathbf{X}$. Thereby for any given set of input locations $\mathbf{X}_*$ we can compute the posterior predictive distribution of the corresponding function values $\mathbf{f}_*$ which is a multivariate normal distribution. Note that instead of the assumption of i.i.d. Gaussian noise, we could also describe dependencies between errors by substituting $\sigma_\mathfrak{n}^2\mathbf{I}$ by any (full rank) covariance matrix of errors in eq. (3.10), e.g. by using a covariance function of errors.

3.2.1. Model Selection

So far we have described inference over the latent function f for a given noise variance $\boldsymbol{\theta} = \sigma_\mathfrak{n}^2$ and hyper-parameters $\boldsymbol{\psi}$ of the Gaussian process prior. Typically, values of these parameters are not known *a priori*. In a full Bayesian setting one should also perform inference over these parameters. Therefore one has to assign prior distributions $p(\boldsymbol{\theta})$ and $p(\boldsymbol{\psi})$ and compute the joint posterior distribution

$$p(\mathbf{f}, \boldsymbol{\theta}, \boldsymbol{\psi}|\mathcal{D}) \;\propto\; p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})\, p(\mathbf{f}|\mathbf{X}, \boldsymbol{\psi})\, p(\boldsymbol{\theta})\, p(\boldsymbol{\psi}) \tag{3.13}$$

of all uncertain elements of the model. Furthermore, the predictions of the model should be averaged over the posterior uncertainty

$$p(\mathbf{f}_*|\mathcal{D}, \mathbf{X}_*) \;=\; \int p(\mathbf{f}_*|\mathbf{f}, \mathbf{X}, \mathbf{X}_*, \boldsymbol{\psi})\, p(\mathbf{f}, \boldsymbol{\theta}, \boldsymbol{\psi}|\mathcal{D})\, d\mathbf{f}\, d\boldsymbol{\theta}\, d\boldsymbol{\psi}\,. \tag{3.14}$$

However, Bayesian inference over all uncertain elements of the model is analytically intractable, because we do not know how to normalise the posterior (3.13) or to compute the integral in eq. (3.14). As will be shown in Section 4.3, the integral (3.14) can be approximated using Markov chain Monte Carlo techniques.

In the regression model with normal noise inference over $\mathbf{f}$ is analytically tractable for fixed values of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$, as shown in the previous section. Instead of doing proper Bayesian inference over $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$, a computationally convenient procedure is to fix them to their respective maximum-likelihood II point estimates as described in Section 2.3.2. Following the ML-II scheme, values for the parameters are found by maximising the conditional evidence

$$(\boldsymbol{\theta}^\star, \boldsymbol{\psi}^\star) \;=\; \operatorname*{argmax}_{\boldsymbol{\theta}, \boldsymbol{\psi}} p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi})\,. \tag{3.15}$$

where $\mathbf{f}$ has been integrated out. The integration over $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ in (3.14) is approximated by fixing their values to their respective ML-II estimates

$$p(\mathbf{f}_*|\mathcal{D}, \mathbf{X}_*) \;\approx\; p(\mathbf{f}_*|\mathcal{D}, \mathbf{X}_*, \boldsymbol{\theta}^\star, \boldsymbol{\psi}^\star) \tag{3.16}$$

such that $\mathbf{f}$ can be integrated out analytically, as in eq. (3.5).

Within the Gaussian process framework the logarithm of the conditional evidence[4] is

$$\ln p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) \;=\; \ln \int p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})\, p(\mathbf{f}|\mathbf{X}, \boldsymbol{\psi})\, d\mathbf{f} \tag{3.17}$$

which can be computed analytically for the regression model with Gaussian noise:

$$\begin{aligned}
\ln p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) &= \ln \int \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma_{\mathrm{n}}^2\mathbf{I})\, \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})\, d\mathbf{f} && \text{(3.18a)}\\
&= \ln \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K} + \sigma_{\mathrm{n}}^2\mathbf{I}) && \text{(3.18b)}\\
&= -\tfrac{m}{2}\ln(2\pi) - \tfrac{1}{2}\ln\left|\mathbf{K} + \sigma_{\mathrm{n}}^2\mathbf{I}\right| - \tfrac{1}{2}\mathbf{y}^\top(\mathbf{K} + \sigma_{\mathrm{n}}^2\mathbf{I})^{-1}\mathbf{y} && \text{(3.18c)}
\end{aligned}$$

using integral (B.24). The log evidence is maximised in $\boldsymbol{\theta} = \sigma_{\mathrm{n}}^2$ and $\boldsymbol{\psi}$ to find point estimates of these parameters given the observed data (see Appendix A.1 for details of a numerically stable implementation). Note that finding the ML-II estimates of the parameters involves a non-convex optimisation problem. Often the parameters are constrained to be positive, e.g. the noise variance σ_{n}^2. Re-parameterising the evidence in terms of $\ln\boldsymbol{\theta}$ and $\ln\boldsymbol{\psi}$ we can make use of standard methods for unconstrained optimisation, e.g. conjugate gradient, to find a local maximum.

Figure 3.2 gives an example of Gaussian process regression with normal noise and illustrates ML-II parameter estimation. The data is generated by sampling 15 x-locations uniformly from $[-10, 10]$ and computing the outputs using the sinc function plus an additive normal noise with standard deviation $\sigma = 0.1$. The GP prior has zero mean and covariance function $k(x, x') = \sigma_s^2 \exp(-\left\|x - x'\right\|^2/(2\ell^2))$ such that $\boldsymbol{\psi} = [\sigma_s^2, \ell]$. The two columns illustrate inference over the latent function f for two different values of ℓ and σ_{n}^2. For the left column some randomly chosen parameter values were used, while the right column shows the posterior process for ML-II estimated parameters. The top row depicts the data and the respective posterior Gaussian processes. Note that in the left plot the resulting posterior process gives a poor fit to the generating function. We now pick two points x_i and x_j that are marked by circles. The middle row shows the contours of the prior and the likelihood in the plane spanned by $f(x_i)$ and $f(x_j)$. The prior is a multivariate normal distribution centred at zero with covariance $k(x_i, x_j)$. The likelihood is a spherical Gaussian centred at $[y_i, y_j]^\top$ with covariance $\sigma_{\mathrm{n}}^2\mathbf{I}$. The lower row shows the corresponding posterior distributions on $f(x_i)$ and $f(x_j)$ which is again a multivariate normal with mean $[m_*(x_i), m_*(x_j)]^\top$ and covariance given element-wise by eq. (3.12b). As described in Section 2.3.1 the evidence quantifies the expectedness of the data, i.e. the agreement between prior and likelihood given the data. Therefore, ML-II estimation gives the parameter values such that prior and likelihood show maximum agreement. The plots in the middle row show that the likelihood peaks sharply and under the ML-II prior $f(x_i)$ and $f(x_j)$ are strongly correlated. The values of the evidence (3.18) for the two settings differ by several orders of magnitude.

Maximum likelihood estimation of parameters is based on the assumption that the

[4]In accordance with Section 2.3 the evidence is denoted by $p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi})$ although technically the evidence for Gaussian process models $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\psi})$ is conditional on $\mathbf{X}$ since we always consider $y|\mathbf{x}$ only.

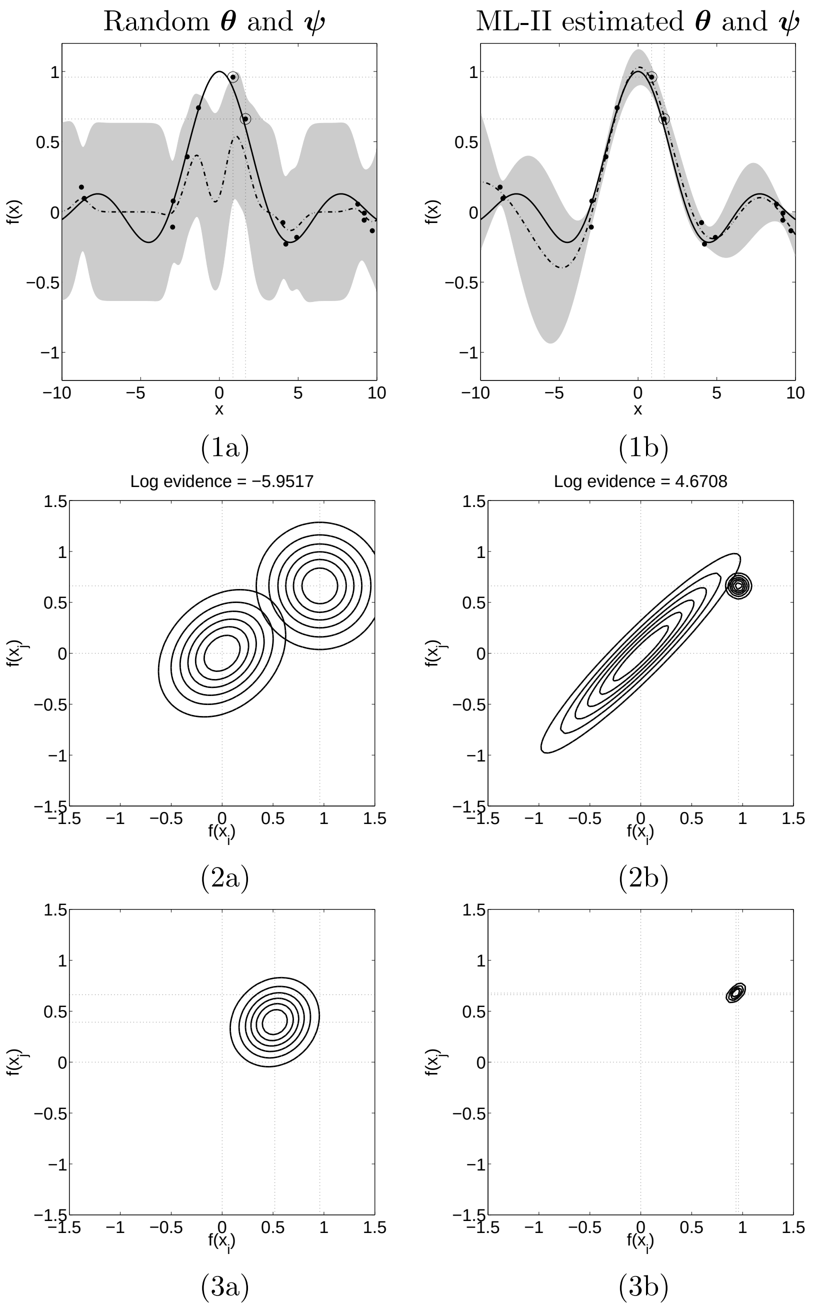


Figure 3.2.: Regression with normal noise and an illustration of ML-II parameter estimation. The data is generated from a sinc function $\operatorname{sinc}(x) = \sin(x)/x$ and additive normal noise with standard deviation $\sigma_\mathrm{n} = 0.1$. The input locations are uniformly sampled from $[-10, 10]$. Column (a) illustrates inference over f given some random values of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ while column (b) is for ML-II estimated values. The top row shows the respective posterior Gaussian processes where the dashed lines describes the mean function and the gray area covers ± 2 standard deviations. Two observations x_i and x_j are selected (marked by circles). The middle row shows the contours of the prior and the likelihood marginalised to the plane spanned by $f(x_i)$ and $f(x_j)$. The lower row shows the corresponding posterior distributions on $f(x_i)$ and $f(x_j)$.

model is correct. However, in any practical data analysis a mismatch between the model and data-generating process seems unavoidable. ML-II type parameter estimation is likely to give reasonable parameters if the discrepancy is somehow small, but it is by no means guaranteed that the ML-II parameters give the most accurate predictive model. In practice it is advisable to compare to other methods of parameter estimation and to assess the generalisation accuracy as a sanity check for model mismatch. This is usually done by using some form of cross-validation, in which the data is repeatedly divided into training and test partitions (Stone, 1974), see also the discussion by Wahba (1990, ch. 4).

Geisser and Eddy (1979) proposed a framework for model selection and parameter estimation by maximising the leave-one-out predictive probability

$$(\boldsymbol{\theta}^\star, \boldsymbol{\psi}^\star) \;=\; \underset{\boldsymbol{\theta},\boldsymbol{\psi}}{\operatorname{argmax}} \prod_{i=1}^{m} p(y_i|\mathcal{D}^{\setminus i}, \mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\psi})\,, \tag{3.19}$$

where $\mathcal{D}^{\setminus i}$ is obtained by removing the ith example $(y_i, \mathbf{x}_i)$ from $\mathcal{D}$. Comparing this criterion to the evidence, especially as it is written in eq. (2.10), we observe that eq. (3.19) directly maximises an estimate of the predictive performance, without measuring the agreement between the model and the data, i.e. the expectedness of the data. Thereby the leave-one-out predictive probability can be expected to give estimates that are more robust with respect to model mismatch, see also the examples given by Rasmussen and Williams (2006, ch. 5.4.3). Sundararajan and Keerthi (2001) show that for the Gaussian process regression model with normal noise the leave-one-out predictive probability can be computed in closed form. Furthermore, the gradient of this quantity can be computed such that gradient based optimisation methods can be used to find the parameter values which maximise this criterion on the available data.

3.2.2. Preprocessing of Data and Nonzero Mean Functions

Above we used a Gaussian process prior with zero mean function $m(\mathbf{x}) = 0$. This corresponds to the *a priori* belief that it is equally likely that the latent function has a positive or negative value at any given input position. This assumption might be reasonable if outputs y are transformed, i.e. $\mathbf{y}$ is pre-processed, such that this assumption holds, e.g. subtracting the mean and removal of trends. But note that in general the mean of the data is not necessarily the mean of the process.

A suitable transformation of either outputs y or inputs $\mathbf{x}$ can also be estimated from the data, thereby making the transformation a part of the model. Using a parametric function to transform the data, ML-II estimation can be used to set its parameters such that the transformed data agrees with the Gaussian process model as well as possible, see for instance Snelson et al. (2004).

It is also possible to use a parametric function $m(\mathbf{x})$ as the mean function and to make inference about its parameters. The simplest possible model is a constant mean function $m(\mathbf{x}) = c$, see Figure 3.3 for an example. O'Hagan (1978, 1994, ch. 10.50) proposes a Gaussian process regression model with linear mean function $m(\mathbf{x}) = \boldsymbol{\beta}^\top\mathbf{x}$

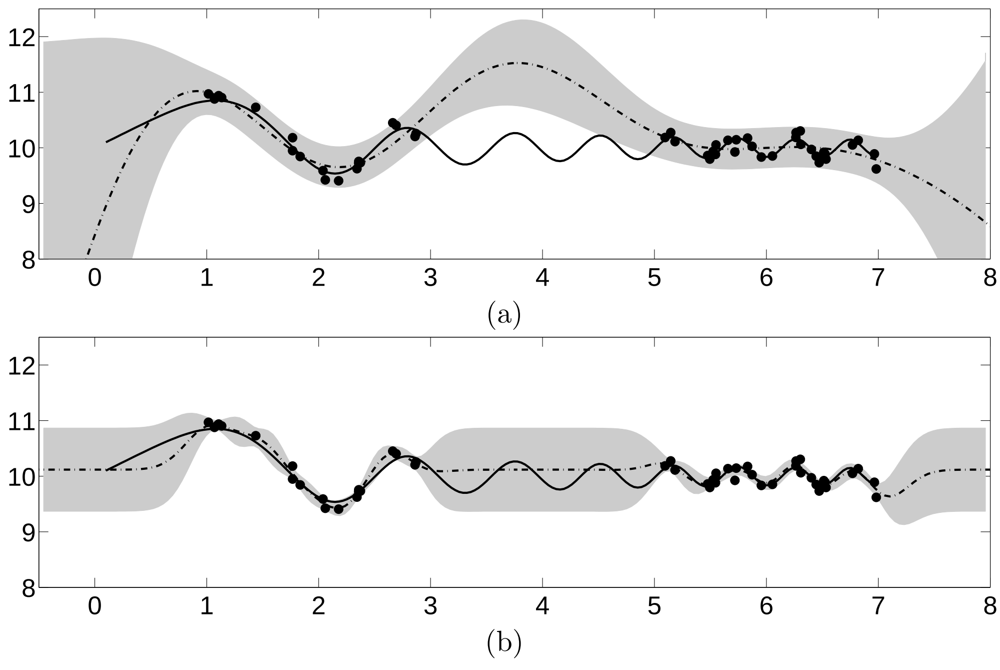


Figure 3.3.: Illustration of the rôle of the mean function in Gaussian process regression. The data was generated from the function shown as solid line plus an additive Gaussian noise. Note that the mean of the data is ≈ 10. For this example a squared exponential covariance function $k(x, x') = \sigma_s^2 \exp(-||x - x'||/\ell^2)$ was used and the signal variance σ_s^2, the characteristic length scale ℓ, and the noise variance σ_n^2 are set to their respective ML-II estimates. Panel (a) shows the posterior GP when the prior mean function is assumed to be zero $m(\mathbf{x}) = 0$. The posterior mean function is given by the dashed line while the grey area depicts two standard deviations. In Panel (b) the mean function of the GP prior was assumed to be constant $m(\mathbf{x}) = c$, and c is set to its ML-II estimate. We observe that the solutions differ, especially in areas where no data has been observed. Given the form of covariance function used, this is not surprising since away from the data the predictive distribution will tend to the prior.

and describes how to make inference over $\boldsymbol{\beta}$ analytically. The prior belief that f has a linear trend can also be modelled by using a covariance function $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}' + \ldots$ that includes a linear term, as will be explained in Section 3.3.4.

3.3. Gaussian Processes & Covariance Functions

As described above a Gaussian process can be used as a non-parametric prior on a latent function f. This section gives some more details on Gaussian processes and covariance functions that will help to understand which characteristics of f are encoded by a GP prior.

3.3.1. Gaussian Processes

A stochastic process f on an index set $\mathcal{X}$ is a random function whose value $f(\mathbf{x})$ is a random variable for any $\mathbf{x} \in \mathcal{X}$. We refer to $\mathcal{X}$ as the input space and unless stated otherwise we assume that $\mathcal{X} = \mathbb{R}^n$. Synonymously a stochastic process is called random field or random function, the latter being the most descriptive term for the use as a prior over functions.

Let $(\Omega, \mathcal{A}, \mathcal{P})$ be a probability space and let $\mathcal{X}$ denote an index set. A stochastic process is a discrete or real valued function $f(\mathbf{x}, \omega)$ which for every fixed $\mathbf{x} \in \mathcal{X}$ is a measurable function of $\omega \in \Omega$.

Usually the dependence on ω will be suppressed but the formal definition is useful to understand why a stochastic process is called a random function. For fixed $\omega \in \Omega$, $f(\mathbf{x}, \omega)$ becomes a non-random function of $\mathbf{x}$. This function is called a realisation or a *sample path* of the stochastic process. For fixed $\mathbf{x} \in \mathcal{X}$ $f(\mathbf{x}, \omega)$ is a random variable on Ω.

The mean of $f(\mathbf{x})$ is defined as $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and the covariance between the two random variables $f(\mathbf{x})$ and $f(\mathbf{x}')$ corresponding to $\mathbf{x}$ and $\mathbf{x}'$ is

$$\operatorname{cov}(f(\mathbf{x}), f(\mathbf{x}')) \;=\; \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \tag{3.20}$$

where the expectation is taken w.r.t. ω. A stochastic process is defined by consistently specifying the finite-dimensional marginal distributions of function values $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)]^\top$ for any collection of inputs $[\mathbf{x}_1, \ldots, \mathbf{x}_m]^\top$.

A Gaussian process is a stochastic process where all such finite-dimensional marginal distributions are multivariate normal

$$\begin{aligned} p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}) &= \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{K}) & (3.21\text{a}) \\ &= (2\pi)^{-\frac{m}{2}} \, |\mathbf{K}|^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}(\mathbf{f} - \mathbf{m})^\top \mathbf{K}^{-1} (\mathbf{f} - \mathbf{m})\right) & (3.21\text{b}) \end{aligned}$$

with mean $\mathbf{m}$ and covariance matrix $\mathbf{K}$. Note that the multivariate normal distribution is completely determined by its first- and second-order moments. Similarly, a Gaussian process is specified by a mean function $m : \mathcal{X} \to \mathbb{R}$ and a positive definite covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{m} = [m(\mathbf{x}_1), \ldots, m(\mathbf{x}_m)]^\top$.

In the above definition the mean function can be chosen arbitrarily but the covariance function must be positive definite in order to ensure the existence of all finite-dimensional distributions. This constraint can be relaxed to positive semi-definiteness of $\mathbf{K}$ in which case we refer to the process as a *degenerate* Gaussian processes. The finite dimensional marginal distributions of a degenerate Gaussian process are *singular multivariate normal.* Let $r = \operatorname{rank}(\mathbf{K})$ and $\lambda_1, \ldots, \lambda_r$ denote the non-zero eigenvalues, then the probability density[5] function of the singular multivariate normal distribution

[5]Note that eq. (3.22) is not a density w.r.t. the Lebesgue measure on $\mathbb{R}^m$ but only on the r-dimensional subspace, see Mardia et al. (1979, p. 41ff) for details.

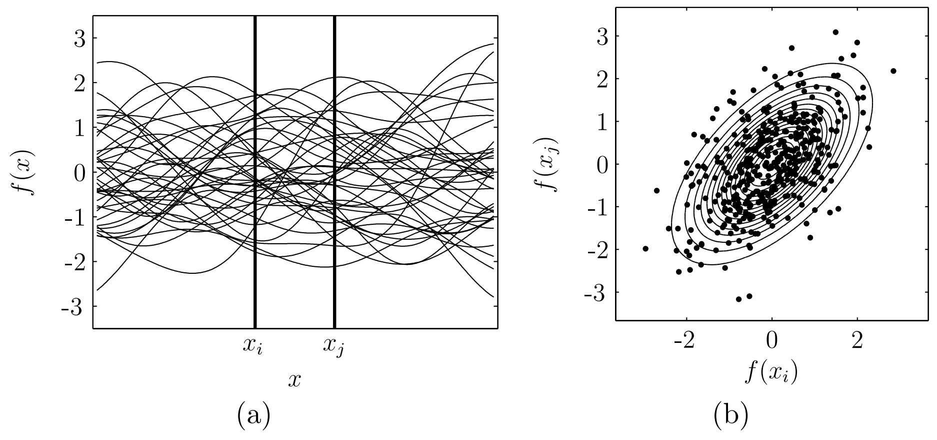


Figure 3.4.: Illustration of the covariance function $k(x, x') = \mathrm{cov}(f(x), f(x'))$ of a Gaussian process. Panel (a) shows 40 sample paths from a zero-mean Gaussian process with covariance function $k(x, x') = \exp(-\|x - x'\|^2/2)$. The vertical lines mark two x-positions x_i and x_j. In panel (b) the values of $f(x_i)$ and $f(x_j)$ for 400 samples are plotted against each other. The contour-lines describe a zero mean bivariate normal density with covariance $k(x_i, x_j)$. Note that the empirical covariance of the samples agrees with the covariance given element-wise by $k(x_i, x_j)$.

can be defined as

$$\mathcal{N}_r(\mathbf{f}|\mathbf{m}, \mathbf{K}) = (2\pi)^{-\frac{r}{2}} \prod_{i=1}^{r} \lambda_i^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}(\mathbf{f} - \mathbf{m})^\top \mathbf{K}^+ (\mathbf{f} - \mathbf{m})\right) \tag{3.22}$$

where $\mathbf{K}^+$ is the pseudo-inverse of $\mathbf{K}$, see Appendix B.1.2.

Sample paths of Gaussian processes cannot be obtained in closed form. For illustration purposes a dense regular grid of $\mathbf{x}$-values (in one or two dimensions) is generated. Then a sample of $\mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{K})$ is plotted against the corresponding $\mathbf{x}$ values and linearly interpolated between the discrete samples (see for example Figure 3.4(a)).

3.3.2. Covariance Functions

The covariance function characterises the dependencies between function values

$$k(\mathbf{x}, \mathbf{x}') = \mathrm{cov}(f(\mathbf{x}), f(\mathbf{x}')) \tag{3.23}$$

for any pair of inputs $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ as illustrated in Figure 3.4. Intuitively, the covariance function $k(\mathbf{x}, \mathbf{x}')$ describes the strength and direction of linear dependence, the similarity, or the mutual informativeness of the function values $f(\mathbf{x})$ and $f(\mathbf{x}')$ as a function of the corresponding inputs.

For k to be a valid covariance function it has to be a symmetric positive semi-definite function. A function is said to be positive semi-definite on $\mathcal{X} \times \mathcal{X}$ if

$$\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \;=\; \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \;\geq\; 0 \tag{3.24}$$

holds for any choice of m, $\boldsymbol{\alpha} \in \mathbb{R}^m$, and $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{X}$. Many covariance functions can be written in the form

$$k(\mathbf{x}, \mathbf{x}') \;=\; \sigma_s^2 c(\mathbf{x}, \mathbf{x}') \tag{3.25}$$

where $|c(\mathbf{x}, \mathbf{x}')| \leq 1$ (and $c(\mathbf{x}, \mathbf{x}) = 1$) is a positive semi-definite *correlation* function and $\sigma_s^2 > 0$ will be referred to as the *signal variance.*

Assuming $\mathcal{X}$ to be a vector space, a covariance function is called *stationary* if it is invariant under arbitrary translation $\mathbf{t} \in \mathcal{X}$ such that $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} + \mathbf{t}, \mathbf{x}' + \mathbf{t})$. Let $\boldsymbol{\tau} = \mathbf{x} - \mathbf{x}'$ denote the separation vector, then a stationary covariance function can be written as $k(\mathbf{x}, \mathbf{x}') = \sigma_s^2 c(\boldsymbol{\tau})$ where $|c(\boldsymbol{\tau})| \leq 1$ is a stationary correlation function. A Gaussian process is called stationary if its covariance function is stationary and its mean function is constant. Gaussian processes with stationary covariance functions are also called homogeneous.

Furthermore, the class of stationary covariance functions contains the isotropic and anisotropic covariance functions. A covariance function is isotropic if it is a function of the Euclidean distance $\tau = \|\mathbf{x} - \mathbf{x}'\|$ only, such that $k(\mathbf{x}, \mathbf{x}') = \sigma_s^2 c(\tau)$. An anisotropic, or ellipsoidal, covariance functions depends on $\boldsymbol{\tau}$ through a norm $\|\boldsymbol{\tau}\|_{\mathbf{W}}^2 = \boldsymbol{\tau}^\top \mathbf{W}^{-1} \boldsymbol{\tau}$ such that $k(\mathbf{x}, \mathbf{x}') = \sigma_s^2 c(\|\boldsymbol{\tau}\|_{\mathbf{W}})$ and $\mathbf{W}$ is positive definite. Any anisotropic correlation function $c(\|\boldsymbol{\tau}\|_{\mathbf{W}})$ is positive semi-definite if $c(\tau)$ is positive semi-definite. This equivalence is trivial since an anisotropic covariance function can be seen as an isotropic covariance function after linear transformation of the inputs. A stationary correlation function is called separable if $c(\boldsymbol{\tau})$ can be written as the product $c(\boldsymbol{\tau}) = c_1(\tau_1) \cdots c_n(\tau_n)$ of valid correlation functions on $\mathbb{R}$.

In general, it is a difficult problem to show positive-semi-definiteness of a function. But the class of positive semi-definite covariance functions is closed under certain operations such that valid covariance functions can be constructed from known positive semi-definite functions. Let k_1, k_2 be valid covariance functions and scalar $d > 0$ then any of the operations

$$k(\mathbf{x}, \mathbf{x}') \;=\; k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \tag{3.26a}$$

$$k(\mathbf{x}, \mathbf{x}') \;=\; k_1(\mathbf{x}, \mathbf{x}') \cdot k_2(\mathbf{x}, \mathbf{x}') \tag{3.26b}$$

$$k(\mathbf{x}, \mathbf{x}') \;=\; k_1(\mathbf{x}, \mathbf{x}') + d \tag{3.26c}$$

$$k(\mathbf{x}, \mathbf{x}') \;=\; k_1(\mathbf{x}, \mathbf{x}') \cdot d \tag{3.26d}$$

results in a valid covariance function (Stein, 1999, ch. 2.3). Furthermore, the inputs can be embedded by some $h : \mathcal{X} \to \mathcal{Y}$ such that $k(h(\mathbf{x}), h(\mathbf{x}'))$ is a valid covariance function if k is positive semi-definite on $\mathcal{Y} \times \mathcal{Y}$. For example h could be a neural network with fixed parameters or a linear transformation. Note that the covariance function (3.12b) of a posterior Gaussian process in the conjugate regression model is also a valid non-stationary covariance function.

3.3.3. Geometrical Properties of Gaussian Processes

Using a Gaussian process prior can be understood as an expression of the belief that the latent function has the form of a realisation, i.e. a sample path, of a Gaussian process. It is therefore of interest to understand the geometrical properties of Gaussian processes which can be linked to characteristics of the covariance function. This section describes some basic properties without any proofs, which can be found in Adler (1981) and Abrahamsen (1997). What is informally referred to as smoothness can be related to continuity and differentiability.

A Gaussian process $f(\mathbf{x})$ on $\mathbb{R}^n$ with continuous mean function is said to be mean-square continuous at $\mathbf{x}_* \in \mathbb{R}^n$ if

$$\lim_{\mathbf{x}\to\mathbf{x}_*} \mathbb{E}\left[|f(\mathbf{x}) - f(\mathbf{x}_*)|^2\right] \;=\; 0 \tag{3.27}$$

and the whole process $f(\mathbf{x})$ is called everywhere mean-square continuous if it is mean-square continuous for all $\mathbf{x} \in \mathbb{R}^n$. Whether a process is mean-square continuous can be determined by looking at the covariance function as stated in the following theorem (Adler, 1981, p. 26): A Gaussian process is mean-square continuous at $\mathbf{x}_* \in \mathbb{R}^n$ if and only if its covariance function $k(\mathbf{x}, \mathbf{x}')$ is continuous at $\mathbf{x} = \mathbf{x}' = \mathbf{x}_*$. If $k(\mathbf{x}, \mathbf{x}')$ is continuous at all $\mathbf{x} = \mathbf{x}' = \mathbf{x}_* \in \mathbb{R}^n$, then f is everywhere mean-square continuous.

Therefore a stationary Gaussian processes is mean-square continuous if the covariance functions $k(\boldsymbol{\tau})$ is continuous at $\mathbf{0}$. Mean-square continuity does not imply continuous sample paths which is more difficult to assure. Adler (1981, p. 60) states the following theorem: Let f be a zero mean Gaussian process with continuous covariance function over $\mathbf{x} \in \mathbb{R}^n$. Then if, for some $0 < C < \infty$ and $\epsilon > 0$,

$$\mathbb{E}\left[(f(\mathbf{x}) - f(\mathbf{x}'))^2\right] \;\leq\; \frac{C}{|\log \|\mathbf{x} - \mathbf{x}'\||^{1+\epsilon}} \tag{3.28}$$

holds for all $\mathbf{x}, \mathbf{x}' \in I^n$, then f has continuous samples paths over the n-dimensional unit cube I^n with probability one. This states that the expected squared difference between two function values should not grow faster than a certain function of the distance between the corresponding inputs. In terms of a covariance function the above theorem implies

$$k(\mathbf{x}, \mathbf{x}) - 2k(\mathbf{x}, \mathbf{x}') + k(\mathbf{x}', \mathbf{x}') \;\leq\; C\left|\log\left\|\mathbf{x} - \mathbf{x}'\right\|\right|^{-(1+\epsilon)} \tag{3.29}$$

which for isotropic covariance functions $k(\mathbf{x}, \mathbf{x}') = \sigma_s^2 c(\tau)$ reduces to

$$1 - c(\tau) \;\leq\; \frac{C}{2\sigma_s^2}\left|\log \tau\right|^{-(1+\epsilon)} \;. \tag{3.30}$$

Abrahamsen (1997, ch. 2.3.1) shows that this property holds, for instance, for correlation functions of the form $c(\tau) = \exp(-\tau^\nu)$ for $0 < \nu \leq 2$.

Let f be a Gaussian process on $\mathbb{R}^n$ with differentiable mean function $m(\mathbf{x})$ and covariance function k. The partial derivative $\dot{f}_i$ of a Gaussian process can be defined as

$$\dot{f}_i(\mathbf{x}, \omega) \;=\; \frac{\partial \dot{f}(\mathbf{x}, \omega)}{\partial x_i} \tag{3.31}$$

assuming differentiability of the sample paths. A Gaussian process f is said to be mean-square differentiable on $\mathbb{R}^n$ if for every sequence $\{\mathbf{x}_n\}$ which converges $\|\mathbf{x}_n - \mathbf{x}\| \to 0$ as $n \to \infty$ also

$$\mathbb{E}\left[(\dot{f}_i(\mathbf{x}_n) - \dot{f}_i(\mathbf{x}))^2\right] \to 0 \tag{3.32}$$

holds. Whether a Gaussian process with differentiable mean function m is mean-square differentiable depends again on the properties of the covariance function. If the derivative $\partial^2 k(\mathbf{x}, \mathbf{x}')/\partial x_i \partial x_i'$ exists and is finite for all $i = 1, \ldots, n$ at the point $(\mathbf{x}, \mathbf{x})$, then $f(\mathbf{x})$ is mean-square differentiable at $\mathbf{x}$ (Abrahamsen, 1997, ch. 2.3.1). In this case the mean and covariance function of $\dot{f}_i(\mathbf{x})$ are given by

$$\dot{m}_i(\mathbf{x}) = \frac{\partial m(\mathbf{x})}{\partial x_i} \quad \text{and} \quad \dot{k}_i(\mathbf{x}, \mathbf{x}') = \frac{\partial^2 k(\mathbf{x}, \mathbf{x}')}{\partial x_i \partial x_i'} \,. \tag{3.33}$$

The sample paths of f are differentiable if the partial derivatives of the sample paths are continuous (Abrahamsen, 1997).

The relations sketched in this section emphasise the importance of the covariance function which determines important properties of the Gaussian process and its sample paths. When using a Gaussian process as a prior over functions the covariance function encodes prior beliefs about the smoothness of this function. The following section describes several covariance functions that have been used in the Gaussian process literature.

3.3.4. Examples of Covariance Functions

For regression analysis the use of stationary covariance functions is often reasonable since it expresses the belief that inputs which are close in $\mathcal{X}$ should have similar function values independent of their location. On the other hand, if the function is known to have different characteristics in different regions of the input space, a non-stationary covariance function might be more appropriate.

Several covariance functions are unsuited for data analysis like the constant covariance function $k(\mathbf{x}, \mathbf{x}') = \sigma_s^2$ or the white noise covariance function which is constant $k(\mathbf{x}, \mathbf{x}') = \sigma_s^2$ if $\mathbf{x} = \mathbf{x}'$ and $k(\mathbf{x}, \mathbf{x}') = 0$ otherwise. Although these are valid covariance functions they do not take the structure of inputs into account.

The general question is how informative is the function value at $\mathbf{x}$ about the function at $\mathbf{x}'$ and vice versa. For stationary covariance functions the answer depends on the difference vector $\boldsymbol{\tau}$ only and a reasonable assumption is that the covariance between function values decays with distance. To control the rate at which the covariance decays with distance we introduce a characteristic length scale $\ell > 0$ parameter. Intuitively this parameter, also referred to as correlation length or practical range, controls at which distance two observations become approximately independent or mutually uninformative.

For anisotropic covariance functions the matrix $\mathbf{W}$ can be used to describe length scale behaviour. If the characteristic length scale is equal in all directions, then $\mathbf{W} = \ell\mathbf{I}$, being equivalent to the isotropic case. This assumes that all input dimensions have the same length scale and are equally informative about the local behaviour of f. This assumption is often too simplistic and instead it is common to set $\mathbf{W} = \operatorname{diag}(\boldsymbol{\ell})$ where $\boldsymbol{\ell} = [\ell_1, \dots \ell_n]^\top$ is a vector of length scale parameters, one for each input dimension respectively. This construction allows specifying the behaviour of f along each axis separately. If the length scale ℓ_i is set to large values, the covariance becomes practically independent of the ith dimension of the inputs $\mathbf{x}$. When (approximate) Bayesian inference is performed over $\boldsymbol{\ell}$ this can be seen as an instance of automatic relevance determination (ARD) as described by Neal (1996, 1998b), because if the posterior distribution on ℓ_i only supports large values, the corresponding input dimension is effectively ignored. Thereby the posterior scale of ℓ_i automatically determines the relevance of the ith input dimension. More complex parameterisations are possible, e.g. $\mathbf{W} = \mathbf{A}\mathbf{A}^\top + \operatorname{diag}(\boldsymbol{\ell})$ where $\mathbf{A}$ can implement any (low rank) transformation of the inputs.

Many correlation functions have been proposed in the literature of which only a handful seems to be used for practical data analysis—or combinations thereof using eqs. (3.26). The following examples are positive definite on $\mathbb{R}^n$. The covariance functions are stated in their anisotropic form, which can be understood as an isotropic covariance function on linearly transformed inputs.

In the literature on spatial (geo-) statistics the Matérn covariance function, also known as the modified Bessel covariance function, is studied extensively. Variations on its parameterisation exist, but the basic functional form is

$$k(\mathbf{x}, \mathbf{x}') \;=\; \frac{\sigma_s^2}{\Gamma(\nu)2^{\nu-1}}(2\sqrt{\nu}\,\|\boldsymbol{\tau}\|_{\mathbf{W}})^{\nu} K_v(2\sqrt{\nu}\,\|\boldsymbol{\tau}\|_{\mathbf{W}}) \tag{3.34}$$

where Γ denotes the gamma function and K_v is the modified Bessel function of the second kind of order $\nu > 0$. The parameter ν controls the differentiability of the covariance function and several other covariance functions can be derived as special cases for specific values of ν, see Abrahamsen (1997, ch. 3.3) or Stein (1999, ch. 2.10) for details.

The exponential, or Laplace, covariance function has the form

$$k(\mathbf{x}, \mathbf{x}') \;=\; \sigma_s^2 \exp\left(-\tfrac{1}{2}\,\|\boldsymbol{\tau}\|_{\mathbf{W}}\right) \tag{3.35}$$

which is equivalent to a scaled Matérn covariance function for $\nu = 1/2$. Note that the sample paths of a GP with this type of covariance function are continuous, but the GP is not mean-square differentiable, since the derivatives of (3.35) at $\boldsymbol{\tau} = \mathbf{0}$ are not defined. The squared exponential covariance function, also referred to as the Gaussian covariance function:

$$k(\mathbf{x}, \mathbf{x}') \;=\; \sigma_s^2 \exp\left(-\tfrac{1}{2}\,\|\boldsymbol{\tau}\|_{\mathbf{W}}^2\right) \tag{3.36}$$

appears to be the most frequently used covariance function in machine learning contexts. Note that the corresponding sample paths are continuous and the corresponding GP is infinitely often mean-square differentiable. Hence, using a squared exponential covariance function in Gaussian process models corresponds to rather strong beliefs about the smoothness of the latent function. It can also be derived as a special case of the Matérn form in the limit $\nu \to \infty$ (Stein, 1999, ch. 2.10). In later chapters the running example will be the anisotropic squared exponential covariance

$$k(\mathbf{x}, \mathbf{x}') \;=\; \sigma_s^2 \exp\left(-\sum_{d=1}^{n} \frac{\|x_d - x'_d\|^2}{2\ell_d^2} \right) \tag{3.37}$$

with individual length scale parameters $\mathbf{W} = \operatorname{diag}(\boldsymbol{\ell})$ for each input dimension. A related covariance function is the rational quadratic

$$k(\mathbf{x}, \mathbf{x}') \;=\; \sigma_s^2 \left(1 + \frac{1}{2a} \|\boldsymbol{\tau}\|_{\mathbf{W}}^2 \right)^{-a} \tag{3.38}$$

which contains the squared exponential covariance function as a limiting case as $a \to \infty$.

Non-stationary covariance functions are more difficult to work with, since typically it is difficult to align the non-stationarity implemented by the covariance function and non-stationarity of the target function, i.e. how the characteristics of the function change over the input space. Examples of non-stationary covariance functions include the neural network covariance functions by Williams (1998) and the constructions proposed by Paciorek and Schervish (2004) and Schmidt and O'Hagan (2003).

Examples of non-stationary semi-definite covariance functions include the class of polynomial covariance functions

$$k(\mathbf{x}, \mathbf{x}') \;=\; \sigma_s^2 (\mathbf{x}^\top \boldsymbol{\Sigma}\, \mathbf{x}' + c)^a \tag{3.39}$$

where $c \geq 0$, $a \in \mathbb{N}$ is referred to as the degree, and $\boldsymbol{\Sigma}$ is a positive definite matrix. For $a = 1$ the class of polynomial covariance functions includes the linear covariance function $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x}' + c$ as a special case. The linear covariance function is of theoretical interest because a degenerate Gaussian process with linear covariance function puts a prior on linear functions. Therefore, Bayesian linear models can be recovered as a special case of Gaussian process models.

Figure 3.5 shows some covariance functions and sample paths from the corresponding Gaussian processes. The top row shows the Matérn covariance function (3.34) as a function of τ for different values of ν. Although the covariance functions appear relatively similar, the corresponding sample paths show different characteristics. Note the relation between the peakedness (differentiability) at $\tau = 0$ and the roughness of the sample paths. The middle row compares the exponential (3.35), the squared exponential (3.36), and the rational quadratic (3.38) covariance functions. For all stationary covariance functions we observe that especially the behaviour for small values of τ is important for the characteristics of the sample paths. The lower row illustrates the effect

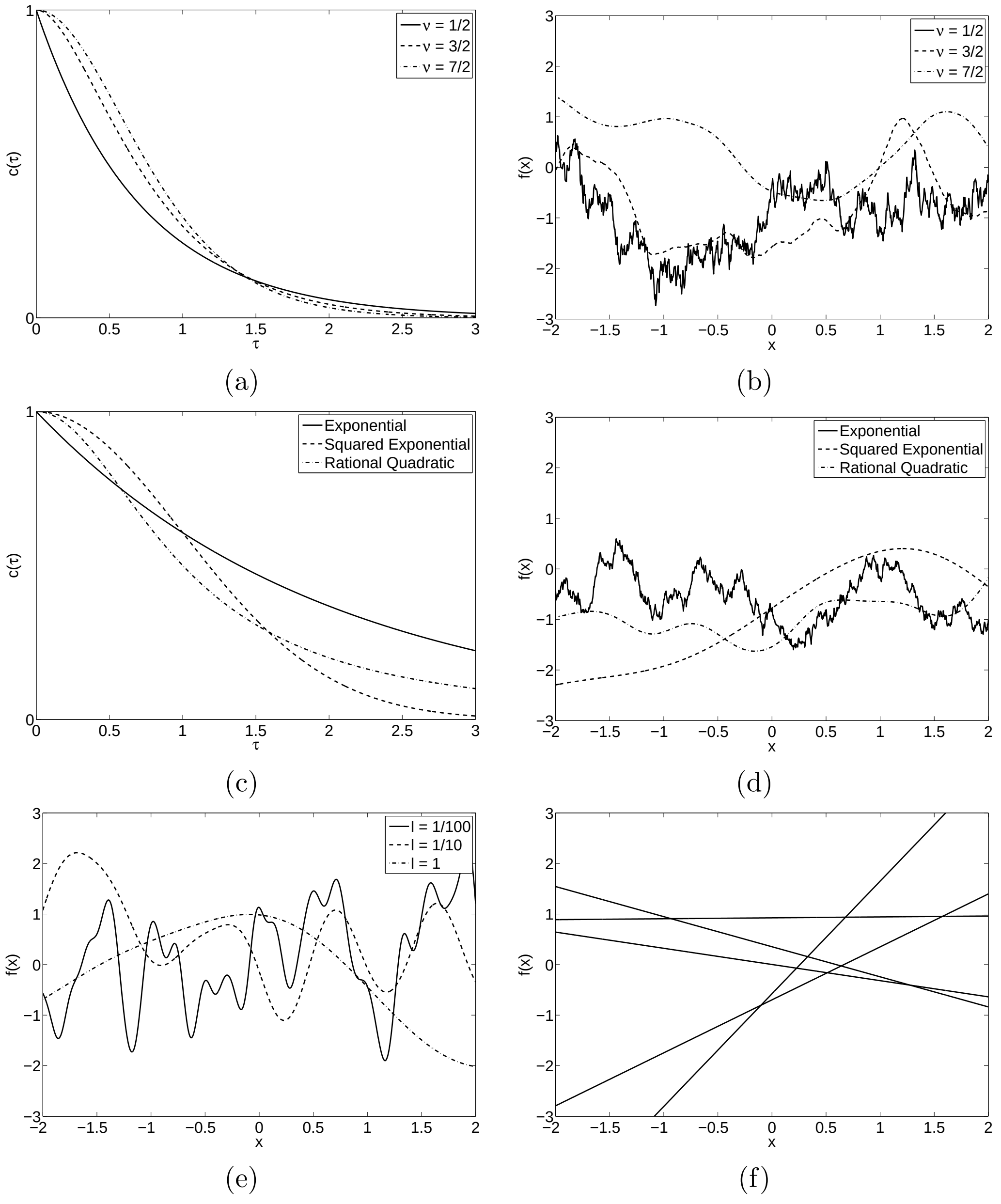


Figure 3.5.: Panel (a) shows the Matérn covariance (3.34) as a function of τ for different values of ν and (b) shows sample paths from the corresponding zero mean Gaussian processes. Panel (c) shows the exponential (3.35), the squared exponential (3.36) and the rational quadratic (3.38) and (d) shows sample paths from corresponding zero-mean Gaussian processes. Panel (e) shows three samples paths of a zero mean Gaussian process with squared exponential covariance function for three different values of the length-scale parameter. Panel (f) shows sample paths from a Gaussian process with linear covariance function (3.39).

of the length scale parameter ℓ exemplified for a squared exponential covariance function (3.37). Intuitively, the characteristic length scale ℓ controls at which distance function values become approximately independent. As an effect, the smaller the characteristic length scale the faster the sample paths can vary.

Figure 3.5(f) shows sample paths of a degenerate Gaussian process with a linear covariance function. This illustrates that the classical linear model can be seen as a particular case of Gaussian process models. In Bayesian linear models the latent function is assumed to be of the form $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ and classically a Gaussian prior $\mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Sigma}_\mathbf{w})$ is used, see for example Box and Tiao (1973). This specifies a prior over linear functions where $\mathbb{E}[f(\mathbf{x})] = 0$ and $\mathrm{cov}(f(\mathbf{x}), f(\mathbf{x}')) = \mathbf{x}^\top \mathbf{\Sigma}_\mathbf{w} \mathbf{x}'$, equivalent to a degernate zero-mean Gaussian process with covariance function $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{\Sigma}_\mathbf{w} \mathbf{x}'$.

3.4. Alternative Interpretations of Gaussian Process Priors

Above we introduced a class of models in which a Gaussian processes is used as a prior over a latent function. We have shown that using a GP corresponds to certain prior beliefs about the latent function, which are expressed by means of the mean and covariance function. However, the Gaussian process prior can also be understood and motivated differently than described above. Two alternative interpretations will be presented briefly in this section. In what is known as the weight-space view, Gaussian processes models are interpreted as Bayesian generalised linear models in a fixed set of basis functions. Furthermore, a Gaussian process prior can be obtained as the limit of an artificial neural network prior as the number of hidden units goes to infinity.

3.4.1. The Weight Space View & Kernel Machines

Assume a function $f(\mathbf{x})$ that comes in the form of a weighted sum of basis functions:

$$f(\mathbf{x}) \;=\; \sum_{i=1}^{N} w_i \phi_i(\mathbf{x}) \;=\; \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) \tag{3.40}$$

where $\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \dots, \phi_N(\mathbf{x})]^\top$ collects the values of the N basis functions at $\mathbf{x}$. If we use a multivariate normal prior $\mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Sigma}_\mathbf{w})$ on the weights, we implicitly specify a prior over functions, since for each realisation of $\mathbf{w}$ sampled from the prior, there is a corresponding function f. Given this form of prior, $f(\mathbf{x})$ for any given $\mathbf{x}$ is a random variable which is distributed according to a normal distribution. Since the mean of $\mathbf{w}$ is zero also the expectation of $f(\mathbf{x})$ is zero. The function values corresponding to any two inputs $\mathbf{x}$ and $\mathbf{x}'$ are jointly Gaussian with covariance (function):

$$\mathrm{cov}(f(\mathbf{x}), f(\mathbf{x}')) \;=\; \boldsymbol{\phi}(\mathbf{x})^\top \mathbb{E}[\mathbf{w}\mathbf{w}^\top] \boldsymbol{\phi}(\mathbf{x}') \;=\; \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{\Sigma}_\mathbf{w} \boldsymbol{\phi}(\mathbf{x}') \;=\; k(\mathbf{x}, \mathbf{x}')\,. \tag{3.41}$$

Therefore, the joint distribution of $m \leq N$ function values $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_m)]^\top$ is multivariate normal with zero mean and a covariance matrix given element-wise by eq. (3.41). Note that this only holds if the resulting covariance matrix is non-singular, which will not be the case if $m > N$, or if the $\boldsymbol{\phi}(\mathbf{x}_i)$ are linearly dependent. In this case $\mathbf{f}$ will follow a degenerate normal distribution (3.22). For finite N this defines a degenerate zero-mean Gaussian process and in the limit of $N \to \infty$ the Gaussian process becomes proper.

This interpretation is known as the *weight-space* view of Gaussian processes while the description used in previous sections has been named the *function-space* view by Williams (1999). Based on the weight-space view it can be argued that a GP prior is equivalent to assuming a linear model on a set of basis functions with a Gaussian prior on the weights. If a covariance function can be written in the form of eq. (3.41) such that $\boldsymbol{\phi}$ is of finite dimensionality N, eq. (3.40) constitutes a parametric model which is equivalent to the corresponding degenerate Gaussian process prior. As an example consider the linear covariance function $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$ which corresponds to $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$ and $\boldsymbol{\Sigma}_\mathbf{w} = \mathbf{I}$, see again Figure 3.5(f).

The weight-space view is closely related to the geometric perspective taken in the non-Bayesian kernel methods. In these methods the kernel function is interpreted as an inner product $k(\mathbf{x}, \mathbf{x}') = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') \rangle_\mathcal{F}$ where $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{F}$ is a mapping into a so called feature space. This interpretation is based on a theorem by Mercer (1909), which states that a positive definite kernel function $k(\mathbf{x}, \mathbf{x}')$ can be written in terms of its orthogonal eigenfunctions ψ_i and corresponding eigenvalues λ_i as

$$k(\mathbf{x}, \mathbf{x}') \;=\; \sum_{i=1}^{N} \lambda_i^2 \, \psi_i(\mathbf{x}) \, \psi_i(\mathbf{x}') \;=\; \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') \rangle_\mathcal{F} \, , \tag{3.42}$$

such that $\boldsymbol{\phi}(\mathbf{x}) = [\lambda_1 \psi_1(\mathbf{x}), \ldots, \lambda_N \psi_N(\mathbf{x})]^\top$, see Schölkopf and Smola (2002, ch. 2.2) for details. For certain kernel functions the mapping $\boldsymbol{\phi}$ can also be written explicitly, e.g. for polynomial kernels (3.39). The basic idea in kernel methods is to reformulate algorithms such that the data only appear in terms of inner products. The linear dot products are then substituted by kernel function evaluations, which are interpreted as dot products of the non-linearly mapped data in some (potentially high dimensional) feature space $\mathcal{F}$.

Note that the covariance (3.41) can be interpreted as a special case of the right hand side of eq. (3.42). Therefore, using the weight space perspective, Gaussian process models can be interpreted as Bayesian (generalised) linear models in a kernel feature space $\mathcal{F}$ with a multivariate normal prior on the weights $\mathbf{w}$. Williams (1999) for instance has shown that the conjugate Gaussian process regression model, as described in Section 3.2, is recovered when implementing Bayesian inference about the weights $\mathbf{w}$ in the basis function representation (3.40). In later chapters only the function-space view will be used, which—from the author's point of view—is the more elegant and practical perspective for Gaussian process models since it avoids the explicit representation of weights $\mathbf{w}$.

3.4.2. Infinite Neural Networks

Neal (1996, ch. 2) describes how a Gaussian process can be obtained as a limit of an

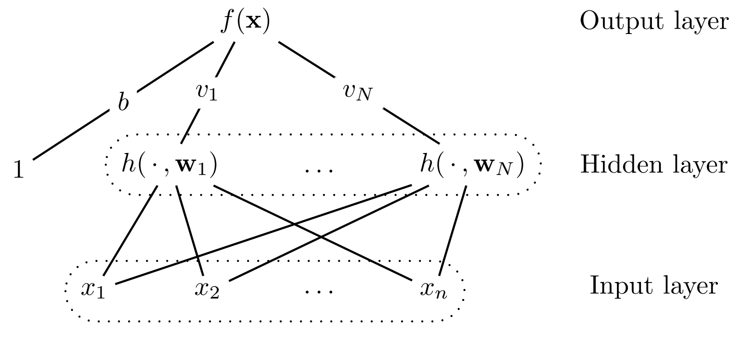


Figure 3.6.: Illustration of an feed-foreward neural network (3.43) with one hidden layer.

artificial neural network with a certain form of priors on the network weights. A one hidden-layer feed-forward neural-network can be written as

$$f(\mathbf{x}) \;=\; b + \sum_{i=1}^{N} v_i \, h(\mathbf{x}, \mathbf{w}_i) \tag{3.43}$$

where N is the number of hidden units, b is a bias term, $h(\cdot,\cdot)$ denotes a bounded transfer function, also referred to as the activation function, and $\mathbf{v}$ and $\mathbf{w}_i$ are parameters, see Figure 3.6 for an illustration. Under certain conditions Hornik et al. (1989) have shown that neural networks of this form are universal approximators, i.e. they are capable of representing any continuous function, as the number of hidden units N tends to infinity.

Again, by defining prior distributions on b, $\mathbf{v}$, and the $\mathbf{w}_i$ we implicitly define a prior over functions. Let $\mathcal{N}(b|0,\sigma_b^2)$ and $\mathcal{N}(\mathbf{v}|\mathbf{0},\sigma_v^2\mathbf{I})$ be the prior distributions on b and the elements of $\mathbf{v}$ respectively. Using this kind of prior the expectation of $f(\mathbf{x})$ is zero

$$\mathbb{E}[f(\mathbf{x})] \;=\; \mathbb{E}[b] + \sum_{i=1}^{N} \mathbb{E}[v_i] \; \mathbb{E}[h(\mathbf{x}, \mathbf{w}_i)] \;=\; 0\,, \tag{3.44}$$

because the expectations of b and v_i are zero and the parameters are independent under the prior. For the variance of $f(\mathbf{x})$ one obtains

$$\mathbb{E}[f(\mathbf{x})^2] \;=\; \mathbb{E}[b^2] + \sum_{i=1}^{N} \mathbb{E}[v_i^2] \; \mathbb{E}[h(\mathbf{x}, \mathbf{w}_i)^2] \;=\; \sigma_b^2 + N\sigma_v^2 \, \mathbb{E}[h(\mathbf{x}, \mathbf{w})^2]\,, \tag{3.45}$$

were we have used that all N transfer functions and the prior distributions of all $\mathbf{w}_i$ are identical. The expectation $\mathbb{E}[h(\mathbf{x}, \mathbf{w})^2]$ is finite since h was assumed to be bounded.

Following Neal (1996, ch. 2) we now let the number of hidden units N go to infinity and simultaneously let the prior variance of the hidden-to-output weights v_i scale as $\sigma_v^2 = \omega^2 N^{-1}$. Given the assumed prior structure the output $f(\mathbf{x})$ of the neural network (3.43) is a sum of $N+1$ independent and identically distributed (i.i.d.) random variables with zero mean and finite variance. In the limit of $N \to \infty$ the central limit theorem states that $f(\mathbf{x})$ becomes normally distributed with zero mean and variance $\sigma_b^2 + \omega^2\, \mathbb{E}[h(\mathbf{x},\mathbf{w})^2]$, see for example Schervish (1997, Theorem B.97). Note that we assumed a particular form of prior on b and $\mathbf{v}$, but the derivation also holds for any other prior with zero mean, finite second moments and independence between parameters.

We now look at the joint distribution of function values $f(\mathbf{x})$ and $f(\mathbf{x}')$. For finite N the covariance of function values at $\mathbf{x}$ and $\mathbf{x}'$ is

$$\begin{aligned} \operatorname{cov}(f(\mathbf{x}), f(\mathbf{x}')) &= \mathbb{E}[b^2] + \sum_{i=1}^{N} \mathbb{E}[v_i^2]\, \mathbb{E}[h(\mathbf{x},\mathbf{w}_i)h(\mathbf{x}',\mathbf{w}_i)] \\ &= \sigma_b^2 + N\sigma_v^2\, \mathbb{E}[h(\mathbf{x},\mathbf{w})h(\mathbf{x}',\mathbf{w})] \;=\; \sigma_b^2 + N\sigma_v^2 k(\mathbf{x},\mathbf{x}') \end{aligned} \tag{3.46}$$

where we have set $k(\mathbf{x},\mathbf{x}') = \mathbb{E}[h(\mathbf{x},\mathbf{w})h(\mathbf{x}',\mathbf{w})]$. Using a multivariate form of the central limit theorem, see for example Schervish (1997, Theorem B.99), an analogous argumentation to the above can be made for the joint distribution of $f(\mathbf{x})$ and $f(\mathbf{x}')$ in the limit as $N \to \infty$. In this limit we obtain that the joint distribution of any collection of function values becomes multivariate normal with zero mean and covariance:

$$\operatorname{cov}(f(\mathbf{x}), f(\mathbf{x}')) \;=\; \sigma_b^2 + \omega^2\, \mathbb{E}[h(\mathbf{x},\mathbf{w})h(\mathbf{x}',\mathbf{w})] \;=\; \sigma_b^2 + \omega^2 k(\mathbf{x},\mathbf{x}')\,. \tag{3.47}$$

Therefore the neural network prior converges to a Gaussian process in the limit of an infinite number of hidden units. Williams (1998) derives the covariance functions corresponding to a sigmoidal and a Gaussian transfer function h.

In the finite case, working with (large) neural networks comes with certain practical difficulties (Hastie et al., 2001, ch. 11.5). For example, it is unclear how to choose an optimal network architecture, which includes the question of the optimal number of hidden units. Another problem is that of local minima and over-fitting in conventional back-propagation maximum likelihood estimation (Bishop, 1995, ch. 4), or multi-modalities in a Bayesian analysis due to symmetries in the architecture. However, by using a Gaussian process—an infinite neural network—prior these problems can often be avoided. In particular, in the conjugate regression model Bayesian inference over f can be done analytically, as shown in Section 3.2.

One could also argue that using a Gaussian processes makes the assumptions about the latent function more transparent and intelligible than a prior over weights in a large neural network, especially in higher dimensional problems when inspecting samples from the prior becomes impractical. In this sense, choosing or designing a covariance function is often easier than specifying priors over network weights such that certain prior beliefs about f are represented. Note that the GP approach also comes with certain disadvantages and limitations, especially when it comes to computational costs of handling large data sets which typically scale as $\mathcal{O}(m^3)$ in time and $\mathcal{O}(m^2)$ in memory, where m refers to the number of data points.

3.5. Bibliographical Remarks

Regression models based on Gaussian processes have been developed in different application contexts in parallel. In spatial statistics, regression with Gaussian processes is known as kriging, named after a South African mining engineer Krige (1951), see for example Cressie (1993) or Stein (1999). A Bayesian description as a non-parametric prior over functions is given by O'Hagan (1978). The noise free model as described in Section 3.1 was used by Sacks et al. (1989) for inference about functions $f(\mathbf{x})$ that are computationally expensive to evaluate. Mardia and Marshall (1984) described an equivalent method to maximum likelihood type II parameter estimation for Gaussian process regression, see also Handcock and Stein (1993). The relation between the posterior mean function and spline techniques has been described by Kimeldorf and Wahba (1970). An empirical comparison between spline models and kriging can be found in Laslett (1994). The degenerate normal distribution can be found in Mardia et al. (1979, p. 41–43).

Neal (1996, ch. 2) described how Gaussian processes can be interpreted as a neural network prior in the limit of an infinite hidden layer. Williams and Rasmussen (1996) introduced Gaussian process regression to the machine learning community. Rasmussen (1996) presented a comparative study of the predictive performance of GP regression (with normal noise) and other non-linear regression techniques, e.g. neural networks. A highly recommendable introduction to Gaussian process models and a summary of GP related developments in the field of machine learning can be found in Rasmussen and Williams (2006). Short introductions are also given by Williams (1999) and MacKay (1998, 2003, ch. 45).

Descriptions of frequentist non-parametric techniques can be found for instance in Härdle et al. (2004), Efromovich (1999), and Györfi et al. (2002) to mention only three. Note that the construction of Gaussian process models is similar to generalised linear models as described by McCullagh and Nelder (1989), but without making the assumption that the latent function is linear.

The presentation of general properties of Gaussian processes and covariance functions is based on the reviews by Abrahamsen (1997), Rasmussen and Williams (2006, ch. 4), and the relevant chapters in Yaglom (1962), Adler (1981), and Stein (1999). The class of covariance functions coincides with the positive semi-definite kernel functions. This class of functions has been studied extensively in the mathematics literature, see for example Schoenberg (1938), Aronszajn (1950), and Saitoh (1988). Positive semi-definite kernel functions gave its name to kernel methods in machine learning, see e.g. Schölkopf and Smola (2002), Genton (2001), or Herbrich (2002).

4. Approximate Bayesian Inference in Gaussian Process Models

Far better an approximate answer to the right question, which is often vague, than the exact answer to the wrong question, which can always be made precise. — Tukey (1962)

The practising Bayesian is well advised to become friends with as many numerical analysts as possible. — Berger (1985)

The previous chapter described the procedure of Bayesian inference about the latent function in Gaussian process models. Unfortunately the necessary calculations are analytically intractable for all but the conjugate model with normal noise described in Section 3.2. For all other models the posterior (3.4), the predictive distribution (3.5), and the evidence (3.17) cannot be computed analytically, so techniques for approximate inference have to be used. Furthermore, for none of the Gaussian process models Bayesian inference about the likelihood parameters $\boldsymbol{\theta}$ and hyper-parameters $\boldsymbol{\psi}$ can be done in closed form.

The basic ingredients of a Gaussian process model are a sampling distribution, i.e. a likelihood, $p(y|f(\mathbf{x}), \boldsymbol{\theta})$ relating the latent function value to an observable quantity y and a Gaussian process prior on f with covariance function $k(\mathbf{x}, \mathbf{x}')$. For the sake of simplicity we again assume that the prior mean function is zero $m(\mathbf{x}) = 0$, see Section 3.2.2 for a discussion. Let $\ln \mathcal{L}(\mathbf{f}) = \ln p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})$ denote the joint log likelihood of $\mathbf{f}$

$$\ln \mathcal{L}(\mathbf{f}) = \sum_{i=1}^{m} \ln p(y_i|f_i, \boldsymbol{\theta}) \tag{4.1}$$

which factorises since the observations are independent given f. The posterior is proportional to the product of likelihood and prior, but the normalisation constant, i.e. the evidence, cannot be computed analytically. The posterior density can be evaluated up to this normalisation constant and let

$$\begin{aligned} \ln \mathcal{Q}(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi}) &= \ln \left(p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})\, \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})\right) && \text{(4.2a)} \\ &= \ln \mathcal{L}(\mathbf{f}) - \tfrac{1}{2} \ln |\mathbf{K}| - \tfrac{1}{2}\mathbf{f}^\top \mathbf{K}^{-1}\mathbf{f} - \tfrac{m}{2} \ln(2\pi) && \text{(4.2b)} \end{aligned}$$

denote the log unnormalised posterior density function. Below we always assume that the likelihood is differentiable w.r.t. f_i and $\boldsymbol{\theta}$ and that the covariance function is differentiable w.r.t. $\boldsymbol{\psi}$, such that the derivatives of $\ln \mathcal{Q}$ with respect to all these quantities can be computed.

For Gaussian process models approximations are either based on a Gaussian approximation to the posterior

$$p(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) \;\approx\; q(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) \;=\; \mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A}) \tag{4.3}$$

or involve Markov chain Monte Carlo (MCMC) sampling. Note that throughout we use p to denote exact quantities and q for approximations.

A key insight is that a Gaussian approximation to the posterior over $\mathbf{f}$ also implies an approximation of the posterior process $f|\mathcal{D}$ by a Gaussian process. This is in analogy to the regression model with normal noise in which the posterior over $\mathbf{f}$ is a multivariate normal distribution and $f|\mathcal{D}$ is a Gaussian process. By substituting the approximate Gaussian posterior (4.3) into eq. (3.5) and using identity (B.19) gives the approximate posterior predictive distribution:

$$\begin{aligned} p(\mathbf{f}_*|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi},\mathbf{X}_*) \;&\approx\; \int p(\mathbf{f}_*|\mathbf{f},\mathbf{X},\mathbf{X}_*,\boldsymbol{\psi})\,\mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A})\,d\mathbf{f} && \text{(4.4a)}\\ &=\; \mathcal{N}(\mathbf{f}_*|\mathbf{K}_*^\top\mathbf{K}^{-1}\mathbf{m},\mathbf{K}_{**}-\mathbf{K}_*^\top(\mathbf{K}^{-1}-\mathbf{K}^{-1}\mathbf{A}\mathbf{K}^{-1})\mathbf{K}_*) && \text{(4.4b)} \end{aligned}$$

which is again a multivariate normal distribution. Since this holds for all possible sets of test inputs $\mathbf{X}_*$ this defines a Gaussian process with mean and covariance function

$$\begin{aligned} m_*(\mathbf{x}) \;&=\; \mathbf{k}(\mathbf{x})^\top\mathbf{K}^{-1}\,\mathbf{m} && \text{(4.5a)}\\ k_*(\mathbf{x},\mathbf{x}') \;&=\; k(\mathbf{x},\mathbf{x}')-\mathbf{k}(\mathbf{x})^\top(\mathbf{K}^{-1}-\mathbf{K}^{-1}\mathbf{A}\mathbf{K}^{-1})\,\mathbf{k}(\mathbf{x}') && \text{(4.5b)} \end{aligned}$$

where $\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}_1,\mathbf{x}),\ldots,k(\mathbf{x}_m,\mathbf{x})]^\top$ is a vector of prior covariances between $\mathbf{x}$ and the training inputs $\mathbf{X}$, compare to eqs. (3.12).

Several approaches to find the mean $\mathbf{m}$ and covariance $\mathbf{A}$ of the posterior approximation (4.3) have been proposed in the literature. In the following Laplace's method (Section 4.1) and Expectation Propagation (Section 4.2) will be described. Both methods also give an approximation of the evidence so that approximate ML-II hyper-parameter estimation can be implemented.

A different approach to approximate Bayesian inference can be implemented using MCMC sampling techniques. In MCMC methods the posterior is not approximated by another distribution, but samples are generated from the posterior, in particular the joint posterior of $\mathbf{f}$, $\boldsymbol{\theta}$, and $\boldsymbol{\psi}$. Markov chain Monte Carlo techniques for Gaussian process models will be described in Section 4.3.

4.1. Laplace's Method

Laplace's method can be used to find a Gaussian approximation $\mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A})$ to the posterior $p(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi})$ over latent function values for fixed $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. The approximation is found by a second order Taylor expansion:

$$\ln \mathcal{Q}(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) \;\approx\; \ln \mathcal{Q}(\mathbf{m}) - \tfrac{1}{2}(\mathbf{m}-\mathbf{f})^\top\mathbf{A}^{-1}(\mathbf{m}-\mathbf{f}) \tag{4.6}$$

around the (largest) mode of the posterior:

$$\mathbf{m} \;=\; \underset{\mathbf{f}\in\mathbb{R}^m}{\operatorname{argmax}}\; \ln \mathcal{Q}(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi}) \;, \tag{4.7}$$

which is the MAP estimate of $\mathbf{f}$. The mode is unique if $\ln \mathcal{L}$ is concave, i.e. if the likelihood $p(y|f(\mathbf{x}), \boldsymbol{\theta})$ is log-concave in the value of the latent function $f(\mathbf{x})$. If this condition is not satisfied, the posterior can be multimodal and so an approximation by a unimodal multivariate normal distribution can be inappropriate. However, if the posterior distribution has a single dominant mode, Laplace's approximation can still be used in the hope that approximating this mode gives a reasonable representation of the main posterior mass. Another potential disadvantage is the locality of Laplace's method since the whole approximation is determined by local properties of the posterior at $\mathbf{f} = \mathbf{m}$.

Assume for the moment that the posterior is unimodal and let:

$$\nabla_{\mathbf{f}} \ln \mathcal{Q} \;=\; \nabla_{\mathbf{f}} \ln \mathcal{L} - \mathbf{K}^{-1}\mathbf{f} \tag{4.8a}$$

$$\nabla\nabla_{\mathbf{f}} \ln \mathcal{Q} \;=\; \nabla\nabla_{\mathbf{f}} \ln \mathcal{L} - \mathbf{K}^{-1} \tag{4.8b}$$

denote the gradient and the Hessian of the unnormalised log posterior density w.r.t. $\mathbf{f}$. The mode of the posterior (4.7) can be found, e.g., using Newton's method, iterating:

$$\mathbf{f}^{(t+1)} \leftarrow \mathbf{f}^{t} - \left(\nabla\nabla_{\mathbf{f}} \ln \mathcal{Q}|_{\mathbf{f}^t}\right)^{-1} \nabla_{\mathbf{f}} \ln \mathcal{Q}|_{\mathbf{f}^t}, \tag{4.9}$$

which usually converges rapidly to $\mathbf{m}$. Note that if the posterior is not unimodal the Hessian might be indefinite and in this case it is more convenient to use a conjugate gradient method to find a local mode. The posterior covariance matrix:

$$\mathbf{A} \;=\; -\left(\nabla\nabla_{\mathbf{f}} \ln \mathcal{Q}|_{\mathbf{m}}\right)^{-1} \;=\; \left(\mathbf{K}^{-1} + \mathbf{W}\right)^{-1} \tag{4.10}$$

is approximated by the curvature at the mode, equal to the negative inverse Hessian, where $\mathbf{W} = -\nabla\nabla_{\mathbf{f}} \ln \mathcal{L}|_{\mathbf{m}}$ denotes a diagonal matrix of second derivatives of the log likelihood. Note that if the likelihood is log-concave its second derivative must be negative such that the Hessian is guaranteed to be positive semi-definite.

Laplace's method also facilitates an approximation to the evidence:

$$p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) \;=\; \int p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})\, p(\mathbf{f}|\mathbf{X}, \boldsymbol{\psi})\, d\mathbf{f} \;=\; \int \exp(\ln \mathcal{Q}(\mathbf{f}))\, d\mathbf{f}. \tag{4.11}$$

Substituting $\ln \mathcal{Q}(\mathbf{f})$ by its Taylor approximation (4.6) the integral becomes Gaussian and can therefore be solved using identity (B.24). The resulting approximate log evidence is:

$$\ln p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) \;\approx\; \ln q(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) \;=\; \ln \mathcal{Q}(\mathbf{m}) + \tfrac{1}{2} \ln |\mathbf{A}| + \tfrac{m}{2} \ln(2\pi) \tag{4.12}$$

and the derivatives of this quantity w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ can be derived and used for optimisation, e.g. using conjugate gradient methods, in an ML-II type setting. Finding the gradients is complicated by an indirect dependency of the mode $\mathbf{m}$ on $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$, see Appendix A.2 for a derivation and details about numerically stable implementations.

4.2. Expectation Propagation

Minka (2001a) proposed the Expectation Propagation (EP) method which can be applied to Gaussian process models. EP finds a Gaussian approximation $q(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) = \mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A})$ to the posterior $p(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi})$ by moment matching of approximate marginal distributions. The starting point is to impose a factorising structure:

$$\begin{aligned} p(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) &= \frac{\mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K})}{p(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})} \prod_{i=1}^{m} p(y_i|f_i,\boldsymbol{\theta}) && \text{(4.13a)} \\ &\approx \frac{\mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K})}{q(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})} \prod_{i=1}^{m} t(f_i,\mu_i,\sigma_i^2,Z_i) \; = \; q(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) && \text{(4.13b)} \end{aligned}$$

resembling the structure of the prior times the factorising likelihood (3.4) where the terms:

$$t(f_i,\mu_i,\sigma_i^2,Z_i) \; = \; Z_i\,\mathcal{N}(f_i|\mu_i,\sigma_i^2) \qquad (4.14)$$

are called *site functions.* Note that the site functions are approximating the likelihood (which normalises over observations y_i), with a Gaussian in f_i, so we cannot expect the site functions to normalise, hence the explicit term Z_i is necessary. For notational convenience we hide the *site parameters* μ_i, σ_i^2, and Z_i and write $t(f_i)$ instead. From eq. (4.14) the Gaussian approximation $q(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi})$ as given by eq. (4.13b) has mean and covariance:

$$\mathbf{m} = \mathbf{A}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \quad \text{and} \quad \mathbf{A} = (\mathbf{K}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}, \qquad (4.15)$$

where $\boldsymbol{\mu} = [\mu_1,\ldots,\mu_m]^\top$ and $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1^2,\ldots,\sigma_m^2)$ collect site function parameters. The EP algorithm iteratively visits each site function in turn, and adjusts the site parameters to match moments of an approximation to the marginal distributions of the posterior. The kth non-central moment of f_i under the posterior is:

$$\begin{aligned} \mathbb{E}[f_i^k] &= \frac{1}{p(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})} \int f_i^k \, p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta})\, p(\mathbf{f}|\mathbf{X},\boldsymbol{\psi})\, d\mathbf{f} && \text{(4.16a)} \\ &= \frac{1}{p(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})} \int f_i^k \, p(y_i|f_i,\boldsymbol{\theta})\, p_{\setminus i}(f_i)\, df_i && \text{(4.16b)} \end{aligned}$$

where:

$$p_{\setminus i}(f_i) \; = \; \int \prod_{j\neq i} p(y_j|f_j,\boldsymbol{\theta})\, p(\mathbf{f}|\mathbf{X},\boldsymbol{\psi})\, d\mathbf{f}^{\setminus i} \qquad (4.17)$$

is called the *cavity distribution* and $\mathbf{f}^{\setminus i}$ denotes $\mathbf{f}$ without f_i. The marginalisation required to compute the exact cavity distribution is intractable. The key step in the EP algorithm is to replace the intractable exact cavity distribution with a tractable approximation based on the site functions:

$$p_{\backslash i}(f_i) \;\approx\; q_{\backslash i}(f_i) \;=\; \int \prod_{j \neq i} t(f_j)\, p(\mathbf{f}|\mathbf{X}, \boldsymbol{\psi})\, d\mathbf{f}^{\backslash i}. \tag{4.18}$$

The approximate cavity function comes in the form of an unnormalised Gaussian $q_{\backslash i}(f_i) \propto \mathcal{N}(f_i|\mu_{\backslash i}, \sigma^2_{\backslash i})$. Multiplying both sides by $t(f_i)$:

$$q_{\backslash i}(f_i)\, t(f_i) \;=\; \int \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}) \prod_{j=1}^{m} t(f_j)\, d\mathbf{f}^{\backslash i} \;\propto\; \mathcal{N}(f_i|m_i, A_{ii}), \tag{4.19}$$

and using basic Gaussian identities we obtain the parameters:

$$\sigma^2_{\backslash i} \;=\; \left((A_{ii})^{-1} - \sigma_i^{-2}\right)^{-1} \quad \text{and} \quad \mu_{\backslash i} \;=\; \sigma^2_{\backslash i}\left(\frac{m_i}{A_{ii}} - \frac{\mu_i}{\sigma_i^2}\right), \tag{4.20}$$

of the approximate cavity function.

The core idea of EP is to adjust the site parameters μ_i, σ_i^2, and Z_i such that the approximate posterior marginal using the exact likelihood approximates as well as possible the posterior marginal based on the site function:

$$q_{\backslash i}(f_i)\, p(y_i|f_i, \boldsymbol{\theta}) \;\approx\; q_{\backslash i}(f_i)\, t(f_i, \mu_i, \sigma_i^2, Z_i) \tag{4.21}$$

by matching the zeroth, first, and second moments.

Matching of moments minimises Kullback-Leibler divergence $\mathrm{KL}(p\,||\,q)$, as defined in Appendix B.2.3.6. Although the classical KL argument only applies to the first and second (and higher) moments for *normalised* distributions, it seems natural also to match zeroth moments.

Therefore, the zeroth, first, and second non-central moment

$$m_k \;=\; \int f_i^k\, p(y_i|f_i, \boldsymbol{\theta})\, q_{\backslash i}(f_i)\, df_i \;=\; \int f_i^k\, p(y_i|f_i, \boldsymbol{\theta})\, \mathcal{N}(f_i|\mu_{\backslash i}, \sigma^2_{\backslash i})\, df_i \tag{4.22}$$

of the left hand side of eq. (4.21) have to be computed for $k = 0, 1, 2$. This can be implemented using numerical integration techniques, but if the moments can be computed analytically this is usually computationally advantageous. In this case a generic approach is to use the moment generating function

$$M(\lambda) \;=\; \int \exp(\lambda f_i)\, p(y_i|f_i, \boldsymbol{\theta})\, \mathcal{N}(f_i|\mu_{\backslash i}, \sigma^2_{\backslash i})\, df_i \tag{4.23}$$

and differentiating with respect to λ gives the non-central moments:

$$m_0 \;=\; M(0), \quad m_1 \;=\; \frac{1}{m_0}\left.\frac{\partial M}{\partial \lambda}\right|_{\lambda=0}, \text{ and } \quad m_2 \;=\; \frac{1}{m_0}\left.\frac{\partial^2 M}{\partial \lambda^2}\right|_{\lambda=0}, \tag{4.24}$$

see for example DeGroot and Schervish (2002, ch. 4.4). By equating these moments with the right hand side of eq. (4.21) the update equations for the site parameters become:

$$\sigma_i^2 = \left((m_2 - m_1^2)^{-1} - \sigma_{\setminus i}^{-2}\right)^{-1} \tag{4.25a}$$

$$\mu_i = \sigma_i^2 \left(m_1(\sigma_{\setminus i}^{-2} + \sigma_i^{-2}) - \frac{\mu_{\setminus i}}{\sigma_{\setminus i}^2} \right) \tag{4.25b}$$

$$Z_i = m_0 \sqrt{2\pi(\sigma_{\setminus i}^2 + \sigma_i^2)} \exp\left(\frac{(\mu_i - \mu_{\setminus i})^2}{2(\sigma_{\setminus i}^2 + \sigma_i^2)} \right) . \tag{4.25c}$$

Once the values of μ_i and σ_i^2 are updated, the effect on $\mathbf{m}$ and $\mathbf{A}$ has to be computed according to eq. (4.15), which in practice is done using rank-one updates of $\mathbf{A}$.

The EP algorithm iteratively updates the site parameters until convergence. A formal proof of convergence does not exist but for log-concave likelihood functions, i.e. when the posterior is concave, EP usually converges reliably.

Finally the evidence can be approximated from the normalisation of eq. (4.13b):

$$\begin{aligned} \ln p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) \approx \ln q(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) &= \ln \int \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}) \prod_{i=1}^m t(f_i)\, d\mathbf{f} \\ &= \sum_{i=1}^m \ln Z_i - \tfrac{1}{2} \ln |\mathbf{K} + \boldsymbol{\Sigma}| - \tfrac{1}{2} \boldsymbol{\mu}^\top (\mathbf{K} + \boldsymbol{\Sigma})^{-1} \boldsymbol{\mu} - \tfrac{m}{2} \ln(2\pi) \end{aligned} \tag{4.26}$$

and its derivatives can be computed in order to implement ML-II parameter estimation of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. The derivatives and further details on implementing EP for Gaussian process models can be found in Appendix A.3, where also a pseudo-code description is given.

Note that if the moments (4.24) cannot be computed analytically, numerical approximations can be used instead, e.g. Gauss-Hermite quadrature (Golub and Welsch, 1969). In this case the derivatives of the form $\partial \ln Z_i / \partial \theta_j$ cannot be computed analytically and so the approximation (4.26) cannot be used in gradient based ML-II estimation. Instead, a variational approximation to the evidence can be used, which will be described in Section 6.3.4.

In practical applications the EP approximation shows to work and converge better for some likelihood models than for others. Unimodality of the posterior—log concavity of the likelihood—seems to be an important factor. Note that in the update equations (4.25) of the site function parameters we ignored the possibility that updates lead to an invalid, non-positive definite covariance matrix $\mathbf{A}$. In those cases one can either skip the update in the hope that a later update will be valid or dampen (soften) the update using a "learning rate" parameter small enough to obtain a positive definite $\mathbf{A}$. However, in general it is not guaranteed that EP converges and often it is a challenging task to implement EP for a particular likelihood avoiding numerical difficulties.

4.3. Markov Chain Monte Carlo

A Gaussian approximation as made by Laplace's method or Expectation Propagation is computationally convenient but its accuracy is limited and difficult to ascertain. In contrast Markov chain Monte Carlo (MCMC) methods are known to be asymptotically exact in the limit of computationally costly simulations. In the remainder of this section we describe MCMC methods for approximate Bayesian inference in Gaussian process models. Note that the two problems of approximating inference over parameters and approximating the value of the evidence have to be handled separately, of which the latter will be considered in Section 4.3.5 only.

Recall the general notation introduced in Chapter 2 by which $p(\mathcal{D}|\boldsymbol{\phi})$ represents a generic model and $\boldsymbol{\phi}$ are continuous and unconstrained parameters. Some data $\mathcal{D}$ have been observed and the objective is to compute the posterior according to Bayes' rule

$$p(\boldsymbol{\phi}|\mathcal{D}) \;\propto\; p(\mathcal{D}|\boldsymbol{\phi})\,p(\boldsymbol{\phi}) \;=\; \mathcal{Q}(\boldsymbol{\phi}|\mathcal{D})\,. \tag{4.27}$$

A common situation is that we can evaluate the likelihood $p(\mathcal{D}|\boldsymbol{\phi})$ and the prior $p(\boldsymbol{\phi})$ for every possible value of $\boldsymbol{\phi}$ but we cannot compute or work with the posterior analytically. Either the normalising constant of $\mathcal{Q}$, i.e. the evidence, cannot be computed or the resulting posterior is of non-standard form and we would be unable to work with it analytically, e.g. for prediction. Markov chain Monte Carlo methods sidestep these problems by generating samples from the posterior $p(\boldsymbol{\phi}|\mathcal{D})$ using only evaluations of the unnormalised posterior $\mathcal{Q}(\boldsymbol{\phi}|\mathcal{D})$.

The conception is that statistics of the samples can be used to approximate properties of the posterior distribution. In particular, the samples can be used to make Monte Carlo approximations of integrals of the form

$$\int h(\boldsymbol{\phi})\,p(\boldsymbol{\phi}|\mathcal{D})\,d\boldsymbol{\phi} \;\approx\; \frac{1}{T}\sum_{i=1}^{T} h(\boldsymbol{\phi}_i) \tag{4.28}$$

where h is any given function and $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_T$ are samples drawn from the posterior $p(\boldsymbol{\phi}|\mathcal{D})$. However, generating samples from a particular distributions is often non-trivial, especially if $p(\boldsymbol{\phi}|\mathcal{D})$ is of non-standard form, which is typically the case for posterior distributions in Bayesian analysis.

Note that the techniques described in the following are by no means restricted to generating samples of posterior distributions $p(\boldsymbol{\phi}|\mathcal{D})$, but could be used to generate samples of any distribution $p(\boldsymbol{\phi})$. However, since in later chapters we will be mostly interested in generating samples from posterior distributions we stick to this particular case.

4.3.1. Metropolis-Hastings Sampling

A general approach is to simulate a Markov chain in the parameter space $\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots$ such that the stationary distribution of the chain is identical to the posterior distribution. This means that $\boldsymbol{\phi}_N$ becomes an approximately independent sample of the posterior as the length of the sequence $t = 1, \ldots, N$ increases. In practice the chain is generated for a finite length N and the state $\boldsymbol{\phi}_N$ is interpreted as samples of the posterior. The procedure is continued until enough samples are obtained such that the characteristics of the posterior distribution can be well approximated by the generated samples.

One basic technique to construct such a Markov chain is the Metropolis-Hastings method which describes how the consecutive state is found. Assume $\boldsymbol{\phi}_t$ is the current state of the chain. In order to find a valid consecutive state $\boldsymbol{\phi}_{t+1}$ a candidate value $\tilde{\boldsymbol{\phi}}$ is sampled from a proposal distribution $p(\tilde{\boldsymbol{\phi}}|\boldsymbol{\phi}_t)$. The proposal is accepted $\boldsymbol{\phi}_{t+1} \leftarrow \tilde{\boldsymbol{\phi}}$ as the consecutive state of the Markov chain if

$$\frac{\mathcal{Q}(\tilde{\boldsymbol{\phi}}|\mathcal{D})}{\mathcal{Q}(\boldsymbol{\phi}_t|\mathcal{D})} \frac{p(\boldsymbol{\phi}_t|\tilde{\boldsymbol{\phi}})}{p(\tilde{\boldsymbol{\phi}}|\boldsymbol{\phi}_t)} \geq u \tag{4.29}$$

where u is a sample from a uniform distribution on the unit interval. Otherwise the proposal is rejected and $\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t$. The first term in the Metropolis-Hastings rule (4.29) captures whether the proposed state $\tilde{\boldsymbol{\phi}}$ yields a higher posterior density. The second term captures how reversible the transition is. Since $\boldsymbol{\phi}_{t+1}$ only depends on $\boldsymbol{\phi}_t$ and not on the history of previous states, the resulting chain is a Markov chain.

The computational efficiency of an MCMC sampling method depends on how the consecutive state is proposed. While simulating the Markov chain, states that occur close-by in the chain are dependent through the proposal distribution. In the simplest form of Metropolis-Hastings sampling, local perturbations of the current state are proposed by sampling from a parametric proposal distribution, for example a Gaussian distribution $p(\tilde{\boldsymbol{\phi}}|\boldsymbol{\phi}_t) = \mathcal{N}(\tilde{\boldsymbol{\phi}}|\boldsymbol{\phi}_t, \boldsymbol{\Sigma})$ centred at the current state. If the proposal $\tilde{\boldsymbol{\phi}}$ is very close to the current state, it is likely to be accepted because the first term in eq. (4.29) will be close to one. But the Markov chain will take a long time to traverse the support of the posterior distribution and successive states will be highly dependent. If $\tilde{\boldsymbol{\phi}}$ is far away from the current state $\boldsymbol{\phi}_t$, then the proposal is likely to be rejected because it will fall into a low density region if the proposal distribution differs significantly from the posterior distribution.

A good proposal distribution should propose states such that the chain moves quickly in the support of the posterior distribution while the acceptance rate is high enough to ensure computational efficiency. The degree to which the chain moves quickly in the support of the posterior distribution is referred to as mixing of the chain. The better the chain mixes the faster the state becomes approximately independent of previous states.

In principle, Bayesian inference can be approximated arbitrarily exact using MCMC as the number of independent samples from the posterior increases. However, for practical data analysis several issues have to be addressed in order to generate as many approximately independent samples as possible in reasonable time. The most important aspect is to design a sampling scheme, i.e. a proposal mechanism and parameterisation, such that the chains mix as well as possible.

The starting value $\boldsymbol{\phi}_0$ should itself be in the posterior support, e.g. the MAP estimate of $\boldsymbol{\phi}$ or a sample of the prior. The chain will move from the initial state to the typical set of the posterior distribution. The initial phase of the simulation before the chain reaches the equilibrium distribution is called the *burn-in* period. The better the chain mixes and the better the initial state is similar to a posterior sample, the shorter the burn-in period will be. Once the chain has converged to the equilibrium distribution states can be picked from the chain at a certain rate, at which the states are approximately independent samples of the posterior. The length of the simulation depends on the necessary number of samples which is equivalent to the precision of the approximation (4.28). In practice, the available computational resources often limit the number of samples in an MCMC approximation. Instead of simulating one long chain, it can be advantageous to run several shorter chains, especially if the posterior is multimodal and a single chain is unlikely to traverse between modes. In general, MCMC sampling can be parallelised this way, although each individual simulation will have its own burn-in period.

Using MCMC techniques in practice requires some experience in designing a sampling scheme and inspecting the resulting chains in order to assess convergence. The aim is to find a MCMC scheme such that the ratio of the number of approximately independent samples and the corresponding computational effort is maximised. Usually several parameters, e.g. of the proposal mechanism, can be tuned in order to improve the mixing behaviour. This is usually done by inspecting trace plots which show the elements of $\boldsymbol{\phi}_t$ over $t = 1, \ldots, T$, see Figure 4.1 on page 54 for an example. The rate at which the states of the chain become approximately independent samples can be estimated by inspecting the empirical auto-correlation structure of samples. A formal proof of convergence of the chain to the posterior distribution is almost impossible to obtain in the general case. In practice the assessment of convergence is often based on visual inspection of trace plots and empirical auto-correlation coefficients, see the discussion of Kass et al. (1998).

4.3.2. Gibbs Sampling

Markov chain Monte Carlo sampling for a particular posterior distribution can often be implemented using various alternative proposal techniques. Instead of updating the whole state $\boldsymbol{\phi}$ it is often more convenient to divide $\boldsymbol{\phi}$ into c components $\boldsymbol{\phi} = [\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^c]$ and update them successively. This decomposition is for instance used in Gibbs sampling where the components of the state are updated element-wise using proposals from the full conditional distributions $p(\boldsymbol{\phi}^i|\boldsymbol{\phi}_t^{\setminus i}, \mathcal{D})$ for $i = 1, \ldots, c$ in turn. Hence, samples have to be generated from the conditional distributions $\boldsymbol{\phi}_{t+1}^i \sim p(\boldsymbol{\phi}^i|\boldsymbol{\phi}_t^{\setminus i}, \mathcal{D})$ where $\boldsymbol{\phi}^{\setminus i}$ denotes all parameters but the ith component. Gibbs sampling is a parameter free instance of the Metropolis-Hastings method in which the proposed states are always accepted. This can be shown easily by substituting the Gibbs proposal distribution into the Metropolis-Hastings rule (4.29):

$$\frac{p(\boldsymbol{\phi}_{t+1}^i, \boldsymbol{\phi}_t^{\setminus i}|\mathcal{D})}{p(\boldsymbol{\phi}_t|\mathcal{D})} \frac{p(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t+1}^i, \boldsymbol{\phi}_t^{\setminus i}, \mathcal{D})}{p(\boldsymbol{\phi}_{t+1}^i, \boldsymbol{\phi}_t^{\setminus i}|\boldsymbol{\phi}_t, \mathcal{D})} = 1 \; . \tag{4.30}$$

A practical problem with Gibbs sampling is that dependencies between parameters are not taken into account. Components that are strongly correlated cannot be updated in a coordinated way, see MacKay (2003, ch. 29.5) for an illustrative example. As an effect the chain may take relatively long to explore the whole posterior support.

4.3.3. Importance Sampling

Importance sampling is another method to approximate the expectation of a function $h(\boldsymbol{\phi})$ with respect to a probability distribution $p(\boldsymbol{\phi}|\mathcal{D})$ as given by eq. (4.28), but avoids sampling from $p(\boldsymbol{\phi}|\mathcal{D})$ directly. The idea is to use samples of another distribution $q(\boldsymbol{\phi})$ called the importance sampler, from which samples can be obtained more easily. Let $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_T$ denote T samples of $q(\boldsymbol{\phi})$. In order to use the samples from q in a Monte Carlo approximation (4.28) one has to correct for not using samples from the correct distribution p one wants to integrate over. This is done by using importance weights of the form

$$w_i \;=\; p(\boldsymbol{\phi}_i|\mathcal{D})/q(\boldsymbol{\phi}_i) \tag{4.31}$$

which capture the relative importance of each sample. If $q(\boldsymbol{\phi}) \neq 0$ whenever $p(\boldsymbol{\phi}|\mathcal{D}) \neq 0$, the approximation

$$\int h(\boldsymbol{\phi})\, p(\boldsymbol{\phi}|\mathcal{D})\, d\boldsymbol{\phi} \;\approx\; \frac{\sum_{i=1}^{T} w_i h(\boldsymbol{\phi}_i)}{\sum_{i=1}^{T} w_i} \tag{4.32}$$

will become exact in the limit as $T \to \infty$. Note that importance sampling does not require that the densities are normalised, since the normalisation constants of $p(\boldsymbol{\phi}|\mathcal{D})$ and $q(\boldsymbol{\phi})$ cancel in eq. (4.32), e.g. it can be used if we can evaluate the unnormalised posterior $p(\mathcal{D}|\boldsymbol{\phi}_i)p(\boldsymbol{\phi}_i)$ only. For finite T the variance of the importance sampling estimate (4.32) depends on the variance of the weights (4.31). The variance will be smaller the more similar q is to p. For high dimensional p it is usually very difficult to find a suitable importance sampler, especially if p is multimodal (MacKay, 2003, ch. 29). Even if a suitable q distribution can be derived, importance sampling is sensible only if it is (much) easier to sample from q than from p directly. For generating samples of common types of distributions that could be used as importance sampler q see for instance Devroye (1986) or Hörmann et al. (2004).

In the following section two Markov chain Monte Carlo techniques will be described that are particularly suited for approximate inference in Gaussian process models. Two aspects have to be addressed separately. The first is how to make predictions based on samples from the posterior and the second is how to estimate the evidence.

4.3.4. Hybrid Monte Carlo

In the Metropolis-Hastings sampling scheme as described above the proposal distribution was fixed and could not adapt to local properties of the posterior distribution. When derivatives of the unnormalised posterior $\mathcal{Q}$ with respect to $\boldsymbol{\phi}$ can be computed, these provide useful information about the direction in which regions of higher posterior density can be found.

4.3.4.1. Hamiltonian Dynamics as a Proposal Mechanism

The Hybrid Monte Carlo method uses simulations in a fictitious physical system to generate proposal states in a Metropolis-Hastings sampler (Duane et al., 1987). The state $\boldsymbol{\phi}$ is interpreted as the location of particles with momentum $\mathbf{q}$. New states are proposed using a procedure that can be understood as a discrete simulation of Hamiltonian dynamics.

The potential energy of $\boldsymbol{\phi}$ is set to $E(\boldsymbol{\phi}) = -\ln \mathcal{Q}(\boldsymbol{\phi}|\mathcal{D})$. The Hamiltonian of the fictitious system is

$$H(\boldsymbol{\phi}, \mathbf{q}) \;=\; -\ln \mathcal{Q}(\boldsymbol{\phi}|\mathcal{D}) + K(\mathbf{q}) \tag{4.33}$$

where the kinetic energy is set to $K(\mathbf{q}) = \|\mathbf{q}\|^2/2$. Instead of sampling the posterior distribution of $\boldsymbol{\phi}$, one generates samples of the joint distribution:

$$p(\boldsymbol{\phi}, \mathbf{q}) \;\propto\; \exp(-H(\boldsymbol{\phi}, \mathbf{q})) \;=\; p(\boldsymbol{\phi}|\mathcal{D}) \exp(-\|\mathbf{q}\|^2/2) \,. \tag{4.34}$$

Since the joint distribution factorises the marginal distribution of $\boldsymbol{\phi}$ is identical to the posterior $p(\boldsymbol{\phi}|\mathcal{D})$. The key idea is that proposal states in a Metropolis-Hastings sampler are obtained by discrete simulation of the Hamiltonian dynamics of the above system. Surprisingly, the augmentation of the problem by introducing auxiliary variables $\mathbf{q}$ effectively eases sampling of the $\boldsymbol{\phi}$. Starting from state $\boldsymbol{\phi}_t$ the initial momentum $\mathbf{q}$ is drawn from a multivariate normal distribution. Then the system is simulated using the Hamiltonian equations

$$\dot{\boldsymbol{\phi}} \;=\; \frac{\partial H(\boldsymbol{\phi}, \mathbf{q})}{\partial \mathbf{q}} \;=\; \mathbf{q} \tag{4.35a}$$

$$\dot{\mathbf{q}} \;=\; -\frac{\partial H(\boldsymbol{\phi}, \mathbf{q})}{\partial \boldsymbol{\phi}} \;=\; \frac{\partial \ln \mathcal{Q}(\boldsymbol{\phi}|\mathcal{D})}{\partial \boldsymbol{\phi}} \tag{4.35b}$$

where the gradient of the unnormalised posterior equals the change in momentum. The simulation is discretised using the leapfrog method, which ensures reversibility of the proposal (Liu, 2001, ch. 9.3). The discretisation requires to set additional parameters, namely the number of steps l (so called leapfrog steps) and the step sizes $\boldsymbol{\epsilon}$, see Appendix A.4.1.

Assume the system is simulated for a certain time and its current state $(\tilde{\boldsymbol{\phi}}, \tilde{\mathbf{q}})$ is proposed as the new state of the Markov chain. The core idea is that while the system is simulated—ideally—the total energy H remains constant, such that $p(\tilde{\boldsymbol{\phi}}, \tilde{\mathbf{q}})$ equals $p(\boldsymbol{\phi}_t, \mathbf{q}_t)$ and the proposed state should be always accepted according to the Metropolis-Hastings rule (4.29). But due to the discretisation an error is introduced such that the simulation can only be considered a proposal mechanism and the acceptance has to be decided on using the Metropolis-Hastings rule. The product of step sizes $\boldsymbol{\epsilon}$ and the number of steps l gives the maximum difference between the starting point of the simulation and its end point. The further away the proposal can be, the better the chain will mix. The acceptance rate is mainly influenced by the step size $\boldsymbol{\epsilon}$ since the larger the step size the larger the approximation errors in the simulation. See Appendix A.4.1 for a pseudo-code description of the algorithm.

4.3.4.2. Hybrid Monte Carlo for Gaussian Process Models

In the remainder of this section it will be described how Hybrid Monte Carlo sampling can be used for approximate inference in Gaussian process models, see also Neal (1998a). Therefore we switch back to the Gaussian process notation introduced in Chapter 3, such that $\mathbf{f}$ refers to a vector of latent function values, $\boldsymbol{\theta}$ denotes additional parameters of the likelihood, and $\boldsymbol{\psi}$ are hyper-parameters, i.e. parameters of the covariance function of the GP prior. In the previous sections approximate inference was implemented over the conditional posterior of latent function values by approximating

$$p(\mathbf{f}|\boldsymbol{\theta},\boldsymbol{\psi},\mathcal{D}) \;\propto\; p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta})\,p(\mathbf{f}|\mathbf{X},\boldsymbol{\psi}) \tag{4.36}$$

by a multivariate normal distribution. Instead of making inference over the likelihood parameters $\boldsymbol{\theta}$ and the hyper-parameters $\boldsymbol{\psi}$, they were set to their ML-II point estimates. However, when using MCMC methods it is more natural to approximate inference jointly over all unknown parameters, see eq. (3.13), which is also favourable from a principled Bayesian perspective. Assuming differentiability of the likelihood and the prior distributions, samples can be generated using Hybrid Monte Carlo from the joint posterior

$$p(\mathbf{f},\boldsymbol{\theta},\boldsymbol{\psi}|\mathcal{D},\boldsymbol{\xi}) \;\propto\; p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta})\,p(\mathbf{f}|\mathbf{X},\boldsymbol{\psi})\,p(\boldsymbol{\psi}|\boldsymbol{\xi})\,p(\boldsymbol{\theta}|\boldsymbol{\xi}) \;=\; \mathcal{Q}(\mathbf{f},\boldsymbol{\theta},\boldsymbol{\psi}|\mathcal{D},\boldsymbol{\xi}) \tag{4.37}$$

of function values, likelihood parameters, and hyper-parameters. In the former notation this corresponds to $\boldsymbol{\phi} = [\mathbf{f},\boldsymbol{\theta},\boldsymbol{\psi}]$. Since inference is also performed over $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$, prior distributions $p(\boldsymbol{\theta}|\boldsymbol{\xi})$ and $p(\boldsymbol{\psi}|\boldsymbol{\xi})$ must be defined, potentially introducing hyper-hyper-parameters $\boldsymbol{\xi}$. The conjugate regression model with normal noise is again a special case since inference over $\mathbf{f}$ can be handled analytically and therefore only sampling $p(\boldsymbol{\theta},\boldsymbol{\psi}|\mathcal{D},\boldsymbol{\xi})$ is necessary.

In the general case, for implementing Hybrid Monte Carlo sampling the value of the unnormalised log posterior

$$\ln \mathcal{Q}(\mathbf{f},\boldsymbol{\theta},\boldsymbol{\psi}|\mathcal{D},\boldsymbol{\xi}) = \ln p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta}) - \tfrac{1}{2}\ln|\mathbf{K}| - \tfrac{1}{2}\mathbf{f}^\top\mathbf{K}^{-1}\mathbf{f} + \ln p(\boldsymbol{\theta}|\boldsymbol{\xi}) + \ln p(\boldsymbol{\psi}|\boldsymbol{\xi}) \tag{4.38}$$

and its derivatives have to be computed. It is advisable to update $[\mathbf{f},\boldsymbol{\theta}]$ and $\boldsymbol{\psi}$ separately, because of different computational costs: evaluating the unnormalised log posterior $\ln\mathcal{Q}$ for different values of $\boldsymbol{\psi}$ requires expensive re-computations of the inverse and determinant of the covariance matrix $\mathbf{K}$. In contrast, evaluating $\ln\mathcal{Q}$ for different values of $\mathbf{f}$ and $\boldsymbol{\theta}$ is computationally inexpensive, since only the log likelihood, the quadratic form and the log priors need to be re-computed.

Even if the covariance function is non-degenerate, inverting $\mathbf{K}$ is often numerically badly conditioned. A common remedy is to add *jitter*, a small multiple of the identity matrix, to the covariance matrix before inverting it. Furthermore, to ease the sampling task by reducing correlations between the elements of $\mathbf{f}$, a linear transformation into new variables $\mathbf{g} = \mathbf{L}^{-1}\mathbf{f}$ can be helpful, such that $\mathbf{g}$ is *white* w.r.t. $\mathbf{K}$, where $\mathbf{K} = \mathbf{L}\mathbf{L}^\top$ is the Cholesky decomposition of the covariance matrix. Since the re-parameterisation is linear, no Jacobian correction is necessary.

Assume T samples of the joint posterior (4.37) have been generated. The samples can be used to make predictions analogously to eq. (3.5) but now averaging over all unknown parameters:

$$p(\mathbf{f}_*|\mathcal{D}, \mathbf{X}_*, \boldsymbol{\xi}) = \int p(\mathbf{f}_*|\mathbf{f}, \boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{X}_*, \mathcal{D})\, p(\mathbf{f}, \boldsymbol{\theta}, \boldsymbol{\psi}|\mathcal{D}, \boldsymbol{\xi})\, d\mathbf{f}\, d\boldsymbol{\theta}\, d\boldsymbol{\psi} \tag{4.39a}$$

$$\approx \frac{1}{T}\sum_{i=1}^{T} p(\mathbf{f}_*|\mathbf{f}_i, \boldsymbol{\theta}_i, \boldsymbol{\psi}_i, \mathbf{X}_*, \mathcal{D}) \tag{4.39b}$$

where each of the $p(\mathbf{f}_*|\mathbf{f}_i, \boldsymbol{\theta}_i, \boldsymbol{\psi}_i, \mathbf{X}_*, \mathcal{D})$ is a multivariate normal distribution of the form given by eq. (3.7), so the predictive distribution is approximated by a mixture of Gaussians.

4.3.5. Annealed Importance Sampling

As described in the previous section MCMC methods can be used to implement approximate inference over all parameters in a Gaussian process model. However, as described in Section 2.3, for model comparison the evidence is the quantity of central importance. This section describes an MCMC technique which can be used to estimate the evidence

$$p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) = \int p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})\, p(\mathbf{f}|\mathbf{X}, \boldsymbol{\psi})\, d\mathbf{f} \tag{4.40}$$

in Gaussian process models conditioned on fixed values of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. Note that in principle one could apply the technique also to integrate over those parameters but in later chapters only the conditional setting will be considered.

The evidence (4.40) comes in the form of an m dimensional integral where m is the number of data points. Good MCMC estimates of normalisation constants are notoriously difficult to obtain, being equivalent to the free-energy estimation problem in physics (Gelman and Meng, 1998).

An apparently simple approach to obtain an estimate of the evidence (4.40) would be to make a Monte Carlo approximation (4.28) by sampling $\mathbf{f}$ values from the prior and to average the corresponding values of the likelihood $p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})$. But the chance that a prior sample agrees with the likelihood would be very small, especially in high dimensional problems. Most samples would have extremely small likelihood and only occasionally a sample would "hit" the likelihood and yield a large value. The resulting estimate would have large variance and is therefore too unreliable in practice. Another approach would be to use importance sampling, for example with an EP or Laplace approximation of the posterior as the importance sampler. However, typically the resulting importance weights (4.31) show large variances as well, such that more sophisticated techniques have to be used in order to obtain reliable estimates (MacKay, 2003, ch. 29).

In the following it will be described how Annealed Importance Sampling (AIS) as proposed by Neal (2001) can be used to obtain reliable estimates of the log evidence. Instead of solving the integral (4.40) directly, a sequence of easier quantities is computed.

We define:

$$Z_t = \int p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})^{\tau(t)} p(\mathbf{f}|\mathbf{X}, \boldsymbol{\psi})\, d\mathbf{f} \tag{4.41}$$

where $\tau(t)$ is an inverse temperature schedule such that $\tau(0) = 0$ and $\tau(T) = 1$. The trick is to rewrite the evidence as a fraction and expanding it:

$$Z = \frac{Z_T}{Z_0} = \frac{Z_T}{Z_{T-1}} \frac{Z_{T-1}}{Z_{T-2}} \cdots \frac{Z_1}{Z_0}, \tag{4.42}$$

where $Z_0 = 1$ since the prior normalises properly. Each term in eq. (4.42) is approximated by importance sampling with

$$q(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi}, \tau(t)) \propto p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})^{\tau(t)} p(\mathbf{f}|\mathbf{X}, \boldsymbol{\psi}) \tag{4.43}$$

as the importance sampler. This results in

$$\frac{Z_t}{Z_{t-1}} = \int \frac{p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})^{\tau(t)} p(\mathbf{f}|\mathbf{X}, \boldsymbol{\psi})}{p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})^{\tau(t-1)} p(\mathbf{f}|\mathbf{X}, \boldsymbol{\psi})} q(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi}, \tau(t-1))\, d\mathbf{f} \tag{4.44a}$$

$$\approx \frac{1}{S} \sum_{i=1}^{S} p(\mathbf{y}|\mathbf{f}_i, \boldsymbol{\theta})^{\tau(t)-\tau(t-1)} \tag{4.44b}$$

where $\mathbf{f}_i$ are samples of $q(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi}, \tau(t))$ generated by Hybrid Monte Carlo sampling. This leaves the choice of how many samples S and how many temperatures T to use. Using a single sample $S = 1$ and a large number of temperatures, the log of each ratio is:

$$\ln\left(\frac{Z_t}{Z_{t-1}}\right) \approx \big(\tau(t) - \tau(t-1)\big) \ln p(\mathbf{y}|\mathbf{f}_t, \boldsymbol{\theta}) \tag{4.45}$$

where $\mathbf{f}_t$ is the only sample at temperature $\tau(t)$. Combining eq. (4.42) with eq. (4.45) we obtain the final estimate:

$$\ln Z \approx \sum_{t=1}^{T} \ln\left(\frac{Z_t}{Z_{t-1}}\right). \tag{4.46}$$

What has been described so far is known as Thermodynamic Integration, which gives an unbiased estimate in the limit of slow temperature changes. In Annealed Importance Sampling the bias caused by finite temperature schedules is removed by combining multiple estimates $Z_1, \dots Z_R$ obtained by R independent annealing runs by their geometric mean

$$\ln p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) \approx \ln\left(\frac{1}{R}\sum_{i=1}^{R} Z_i\right), \tag{4.47}$$

see Neal (2001) and Appendix A.4.2 for further details.

4.3.6. An Example For Gaussian Process Regression

This section presents examples for Hybrid Monte Carlo and Annealed Importance Sampling used for approximate Bayesian inference in Gaussian process models. Since one of our interests will be to probe the accuracy of AIS, the example will be based on the conjugate Gaussian process regression model with normal noise for which the evidence can be computed analytically. For the experiments the Boston Housing data set is used which has been introduced by Harrison and Rubinfeld (1978) and has become a popular reference problem in non-linear regression. The data will be analysed in more detail in Section 5.6.2. The task is to predict the median price of houses in different parts of the Boston metropolitan area based on $n = 13$ input variables. The data set consists of 506 observations which we normalise to zero mean and unit variance. Then the data is randomly split into a training set of $m = 400$ observations, leaving the remaining 106 observations for testing.

In all experiments an anisotropic squared exponential covariance function (3.37) is used with individual length scale parameters ℓ_i for each input dimension $i = 1, \dots, n$ respectively. In total the model has 14 hyper-parameters $\boldsymbol{\psi} = [\boldsymbol{\ell}, \sigma_s^2]$ and the noise variance $\boldsymbol{\theta} = \sigma_{\mathrm{n}}^2$ is the only likelihood parameter.

4.3.6.1. Hybrid Monte Carlo

Since inference over $\mathbf{f}$ can be done analytically we only need to approximate inference over the noise variance and the hyper-parameters. All parameters are positive and for simplicity we use wide, independent Gaussian priors on the log value of the variables. We use Hybrid Monte Carlo to generate samples of the posterior distribution

$$p(\boldsymbol{\theta}, \boldsymbol{\psi}|\mathcal{D}) \;\propto\; p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi})\, p(\boldsymbol{\theta})\, p(\boldsymbol{\psi}) \tag{4.48}$$

where the log of $p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi})$ is given by eq. (3.18c). The parameters of the leap-frog discretisation were set by visual inspection of chains in some initial simulations.

Figure 4.1 shows trace plots of chains of length $T = 500$ generated by HMC using $l = 100$ leapfrog steps each of size $\epsilon = 0.08$. The burn-in period appears not to be longer than 20 samples. The acceptance rate was approximately 90% and the samples show relatively low auto-correlation. While sampling, some of the length scales show to be better determined by the data than others, i.e. the posterior uncertainty in the value of the respective length scale parameters differs. Note that the larger the characteristic length scale the less influential the corresponding input dimension will be in a prediction. The posterior uncertainty in the signal variance σ_s^2 and the noise variance σ_{n}^2 show to be relatively small.

As a side remark, note that compared to ML-II estimation of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ approximating inference over these parameters using MCMC sampling usually improves the predictive performance significantly. For the given example we discard the first 50 samples as burn-in period and subsequently pick every 10th state as a posterior sample. Averaging over the corresponding predictive distributions, we observe that the root mean square error can be reduced by approximately 10% compared to using just the ML-II estimates.

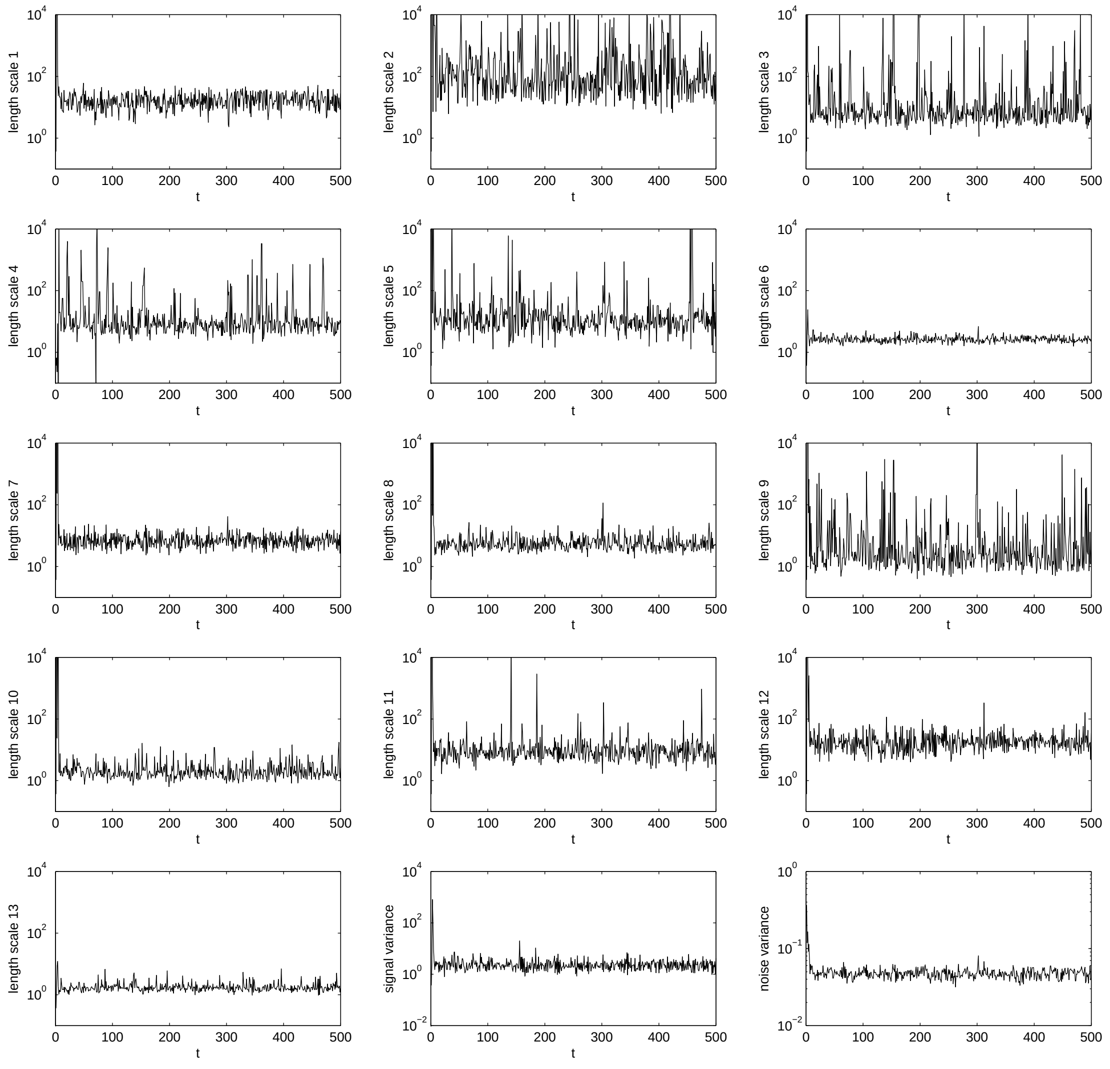


Figure 4.1.: Trace plots for Hybrid Monte Carlo sampling from $p(\boldsymbol{\theta}, \boldsymbol{\psi}|\mathcal{D})$ for Boston Housing data set for $t = 1, \ldots, 500$. For the 13 length scale parameters $\boldsymbol{\ell}$, we observe that the respective chains show different mean values and amplitudes. Note that the larger the value the more the covariance between examples becomes independent of the respective input dimension. The last two plots correspond to the signal variance σ_s^2 (larger values) and the noise variance σ_{n}^2 (lower values). Usually, one would also inspect the potential energy, which is not plotted here, but also does not indicate any noticeable problems, e.g. trends or shifts.

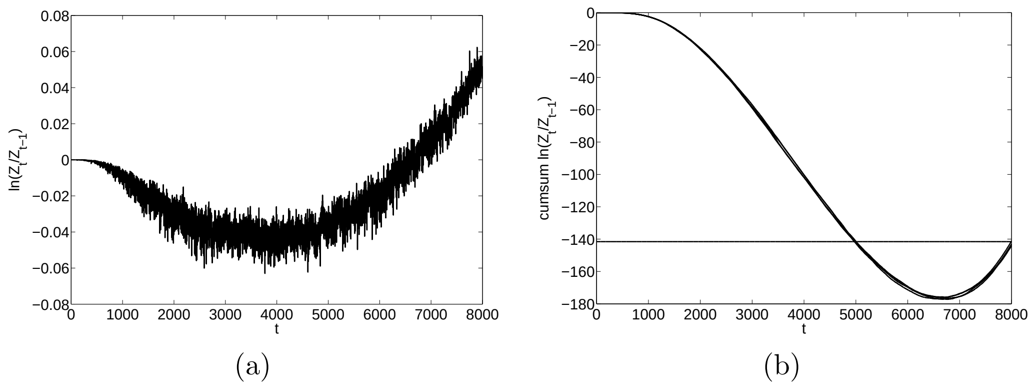


Figure 4.2.: Estimating the evidence using Annealed Importance Sampling. Panel (a) shows the value of $\ln(Z_t/Z_{t-1})$ over t for one run of Thermodynamic Integration. Panel (b) shows the cumulative sums for all $R = 4$ runs. The horizontal line marks the true analytic value.

However, finding ML-II estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ is computationally faster by an order of magnitude, taking only a few minutes compared to a few hours for MCMC sampling. In the above experiment we simulated 500 samples using 100 leapfrog steps which sums to 50000 evaluations of the potential energy, i.e. the log of the right hand side of eq. (4.48), and its gradients. Note that each evaluation requires to compute the log evidence which involves a Cholesky decomposition of an $[m \times m]$ covariance matrix $\mathbf{K}$, see Appendix A.1. In contrast, ML-II parameter estimation using a conjugate gradient optimisation scheme requires on the order of 200 evaluations of the log evidence (3.18c).

4.3.6.2. Annealed Importance Sampling

The purpose of the presented experiment is solely to demonstrate that AIS can be used for estimating the evidence in Gaussian process models. In later chapters AIS will be used to estimate the evidence when it cannot be computed analytically.

For the Boston Housing data described in the previous section, the ML-II estimates of the likelihood parameters $\boldsymbol{\theta}^\star$ and hyper-parameters $\boldsymbol{\psi}^\star$ where found by maximising the log evidence (3.18c). The true value of the evidence conditioned on the ML-II estimates is a $m = 400$ dimensional integral whose analytic solution is $\ln p(\mathcal{D}|\boldsymbol{\theta}^\star, \boldsymbol{\psi}^\star) = -141.55$.

Four runs ($R = 4$) of Thermodynamic Integration were simulated with a temperature schedule $\tau(t) = (t/T)^4$ for $t = 0, \dots, 8000$. Figure 4.2(a) shows the values of $\ln(Z_t/Z_{t-1})$ as given by eq. (4.45) for one of the runs. Figure 4.2(b) shows the cumulative sums of $\ln(Z_t/Z_{t-1})$ over t for all four runs. The horizontal line marks the true value and the endpoints of all four estimates are relatively close by. Combining the runs into an AIS estimate of the log evidence yields a value of -142.23 which is off the true value by less than one log unit.

4.4. Bibliographical Remarks

Laplace's approximation is a standard technique and is described in many introductory texts on Bayesian analysis, e.g. Bernardo and Smith (1994, ch. 5.5.1) and MacKay (2003, ch. 27). Williams (1998) proposed Laplace's method for approximate inference in the Gaussian process model for classification, which will be described in Chapter 6. It was also used for ordinal regression by Chu and Ghahramani (2005).

Expectation Propagation as proposed by Minka (2001a,b) is an extension to online moment matching methods, also known as assumed-density filtering (ADF), as described for example by Lauritzen (1992). The approximations proposed by Opper and Winther (2000) are equivalent to EP.

General references to Markov chain Monte Carlo methods with a focus on Bayesian analysis are Neal (1993), Gilks et al. (1996), and Liu (2001). A recommendable introduction is given by MacKay (1999b). The Metropolis-Hastings method is named after the work of Metropolis et al. (1953) and Hastings (1970). Gibbs sampling is due to Geman and Geman (1984). Hybrid Monte Carlo has been introduced by Duane et al. (1987) and its use in the context of Gaussian process models was proposed by Neal (1997, 1998a). MacKay (2003, ch. 30) proposes the name Hamiltonian Monte Carlo for obvious reasons.

The problem of estimating the evidence using MCMC techniques is reviewed by Gelman and Meng (1998). Annealed Importance Sampling has been proposed by Neal (2001) and it has been shown above that it can be used for approximating the evidence in Gaussian process models. The analysis of convergence of Markov chains is addressed by Gelman (1996) and Cowles and Carlin (1996). Several other important practical issues have been described by Raftery and Lewis (1996), see also the round table discussion by Kass et al. (1998). For details on generating random samples of standard distributions see for example Devroye (1986).

5. Robust Gaussian Process Regression

All models are wrong, but some are useful. — Box (1979)

To solve a real-world regression problem the analyst should carefully screen the data and use all prior information at hand in order to choose an appropriate regression model. However, in practice a discrepancy between the model and the data-generating process seems unavoidable. Robust Bayesian methods can be understood as attempts to limit undesired distractions and distortions that result from this mismatch.

5.1. Bayesian Perspective on Robustness

In the Bayesian framework the notion of robustness is associated with two different aspects of stability. One aspect is the stability of the conclusions with respect to small changes of the prior distribution. This desideratum is related to the practical problem that the prior beliefs can only be formalised up to a limited accuracy. It is therefore important that the conclusions are known to be insensitive to this source of inaccuracy. Another kind of stability is desired with respect to inaccuracies of the sampling distribution, relative to the true data-generating process, i.e. limiting the effects of model mismatch. If observations occur that are highly implausible under the likelihood model, these cannot be rejected and can become decisive in the analysis. Whenever surprising data are observed, i.e. data that is highly unlikely to be generated from the model, sensitivity of posterior inferences must be suspected (O'Hagan, 1994, ch. 7.19). Robust models aim at limiting the influence of such unexpected observations and the effects of model mismatch.

In frequentist methods, a robust statistic is a criterion (such as an estimator) that is insensitive to large errors in small portions of the data (Huber, 1981). Box and Tiao (1964, 1973, ch. 3.2) suggest the terms *inference* robustness and *criterion* robustness to distinguish between the Bayesian and the frequentist notion. In the following we will focus on inference robustness and describe several alternative sampling distributions of errors in Gaussian process regression models.

In the context of regression analysis, robustness is associated with the notion of *outliers*, which refers to observations that deviate strongly from the regular structure. Often the presence of such outliers is attributed to observational errors, e.g. data processing errors or failures of measuring instruments, but it can also be an original feature of the data-generating process. Another possible explanation of large errors could be that not all relevant inputs are available or included in the model. Therefore, if an unconsidered source of variation occasionally affects the target variable, the effect could be interpreted as outliers, because the model cannot distinguish between random noise and systematic

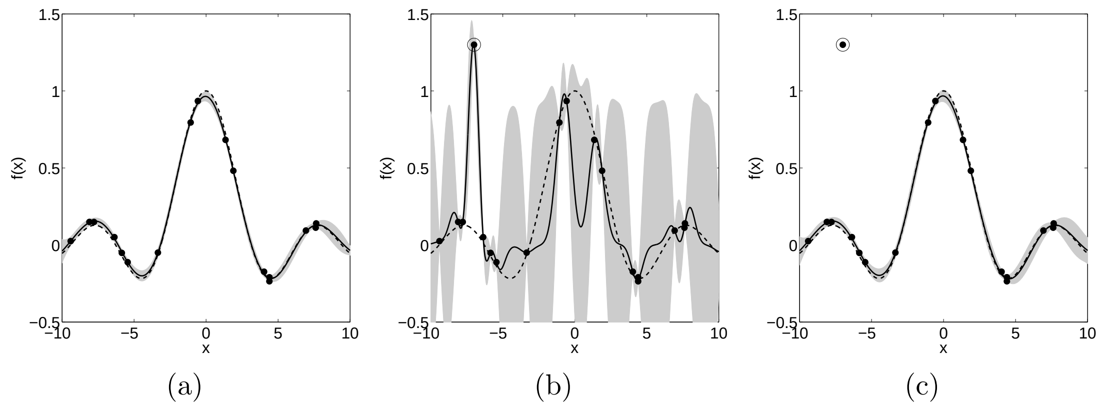


Figure 5.1.: Illustration of the effect of a single outlier in Gaussian process regression models depending on the noise assumption. The dashed line shows the sinc function $f(x) = \sin(x)/x$ and the points mark noisy samples thereof. Panel (a) shows the posterior process of a GP model with normal noise and ML-II estimated parameters. The posterior process is represented by its mean (solid line) and ± 2 standard deviations (shaded area). Panel (b) shows that adding a single outlier (marked by a circle) highly affects the ML-II estimation and the posterior process when the noise is assumed to normal. As before, the mean function roughly interpolates the samples while the uncertainty about the function is increased dramatically. This can be explained as an effect of the inferred shorter length scale ℓ and larger signal variance σ_s^2. Note that this can be seen as an instance of over-fitting, i.e. random fluctuation is interpreted as systematic variation. Panel (c) shows the posterior GP obtained when the noise is modelled as a mixture of two normal distributions as will be described in Section 5.3.

effects due to external dependencies. In this sense, outliers could be understood as an indicator for model mismatch as well.

Decomposing the data-generating process into a systematic and a random component, outliers occur when the random distortion can occasionally be very large. Box and Tiao (1968) describe an outlier as being an observation which is suspected of being partially or wholly irrelevant because it is not generated by the stochastic model assumed or its error is so large that it is effectively uninformative about the systematic component.

As Jaynes (2003, ch. 21) phrases it: "One seeks data analysis methods that are *robust*, which means insensitive to the exact sampling distribution of errors, as it is often stated, insensitive to the model, or are, *resistant*, meaning that large errors in small proportion of the data do not greatly affect the conclusions." So a statistical model can be called robust if it leads to conclusions which are insensitive to the occurrence of outlier observations. Note that this implies that an observation can only be called an outlier relative to a given model. Therefore, when referring to observations as outliers below, the interpretation as outlier must be seen relative to a normal distribution.

5.2. Robust Gaussian Process Regression Models

The Bayesian approach to robust regression, i.e. handling outliers, results automatically from the common statement that a model should be chosen so as to reflect all the analyst's beliefs and uncertainties. So a Bayesian regression model can be considered robust if it explicitly accounts for the potential existence of outliers, i.e. extra-normal variation. Therefore, unless the analyst has absolutely no doubt that a model accounts for all possible observations—in other words, unless one is certain that there *are* no outliers relative to that model—one should adjust the model to account explicitly for the potential occurrence of outliers.

As described in Section 3.1 the Gaussian process regression model assumes a latent function $f(\mathbf{x})$ of which only noisy samples $y = f(\mathbf{x}) + \varepsilon$ can be observed. The error ε is assumed to be independent and identically distributed according to a sampling distribution $p(\varepsilon|\boldsymbol{\theta})$ which is usually a symmetric distribution with zero mean, i.e. $\mathbb{E}[\varepsilon] = 0$.

In Section 3.2 it was described that the assumption of normal noise enables us to compute the posterior process and the evidence in closed form. However, as illustrated in Figure 5.1 when the noise is assumed to be normal, a single outlier can drastically influence the posterior process and in particular the maximum likelihood II estimates of parameters. As described in Section 2.3.2, ML-II estimation can be understood intuitively as finding values of the likelihood parameters and the hyper-parameters of the prior such that the observed data would have been as *unsurprising* as possible. If the model and the data-generating process differ significantly, observations with outliers (relative to the assumed model) will be very surprising and the outliers will be highly influential on the ML-II estimates.

The classical justification for assuming normal noise is based on the central limit theorem. The idea is that the errors are compounded of a very large number of random additive perturbations originating from independent sources. However, in practical data analysis this argument is often not appropriate. In fact, the precise distribution of errors is usually unknown and a model mismatch almost certain. Nevertheless, by using a likelihood that can explain a wider range of errors we can reduce the effects of this mismatch, i.e. a more robust model.

To account for more extreme observations the sampling distribution of errors $p(\varepsilon|\boldsymbol{\theta})$ has to be a more kurtotic distribution, i.e. a leptokurtic distribution with more probability mass in its tails relative to a normal (Box and Tiao, 1973, 3.1.1). In the following several alternative noise models and respective techniques for approximate inference will be discussed. The question which noise model is preferable from a conceptual point is ill posed since the answer will always depend on the data set at hand, i.e. how well the (noise) model matches the data-generating process. Instead, the focus will be to compare the practicability of several alternative approximation schemes.

All noise models studied in the following can be represented as *scale mixtures of normal distributions* as introduced by Andrews and Mallows (1974). These distributions can be written as

$$p(y|f(\mathbf{x}),\boldsymbol{\theta}) \;=\; \int \mathcal{N}(y|f(\mathbf{x}),\sigma^2)\, p_M(\sigma^2|\boldsymbol{\theta})\, d\sigma^2 \tag{5.1}$$

where $p_M(\sigma^2|\boldsymbol{\theta})$ will be referred to as the *scale-mixture distribution.* This class of distributions contains mixtures of finitely many normal distributions with different individual variances as a degenerate case (Section 5.3). The Student-t distribution can be obtained as a mixture of infinitely many normal distributions when the scale-mixture distribution is chosen to be an inverse gamma distribution (Section 5.4). Using an exponential distribution as the scale-mixture distribution gives the Laplace, or double exponential, distribution (Section 5.5).

5.3. Mixture Noise Models

Bayesian regression models can be considered robust if the possible occurrence of outliers is represented explicitly in the sampling distribution of errors. An intuitive approach is to introduce separate noise models for regular and extreme observations respectively. Jaynes (2003, ch. 21) calls it a "two-model model" being a mixture of a model which accounts for the *regular* observations and a second model for explaining *outliers.* The "two-model model" will be the line of thought in the remainder of this section although also mixtures of more than two models are covered.

Let $p_r(y_i|f_i,\boldsymbol{\theta})$ denote a noise model which describes our beliefs about *regular* observations, like the typical error of a measuring instrument. Furthermore, assume that the potential occurrence of outliers cannot be denied. For these outliers we believe the distribution of errors $p_o(y_i|f_i,\boldsymbol{\theta})$ to be different. If we use π to denote the fraction of outlier observations, we can combine both models

$$p(y_i|f_i,\boldsymbol{\theta}) \;=\; (1-\pi)\; p_r(y_i|f_i,\boldsymbol{\theta}) + \pi\; p_o(y_i|f_i,\boldsymbol{\theta}) \tag{5.2}$$

and obtain a mixture likelihood, i.e. a "two-model model". In the following we consider the mixture of two normal distributions as proposed by Box and Tiao (1968). For *regular* observations we assume a relatively small variance σ_r^2 compared to the variance σ_o^2 of the outlier distribution. Thus the Gaussian mixture noise model is

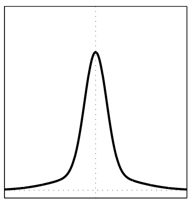


Figure 5.2.: The p.d.f. of a mixture of two Gaussians.

$$p(y_i|f_i,\boldsymbol{\theta}) \;=\; (1-\pi)\,\mathcal{N}(y_i|f_i,\sigma_r^2) + \pi\,\mathcal{N}(y_i|f_i,\sigma_o^2) \tag{5.3}$$

where $\boldsymbol{\theta} = [\pi, \sigma_r^2, \sigma_o^2]$ collects the likelihood parameters. Assuming p_r to be a normal distribution is a common and often plausible hypothesis. It seems more questionable to explain the outliers by normal noise with relatively large variance. If we were certain that this were the case, the Gaussian mixture noise model would be correct and we would not call it robust. Generally, if the outlier-generating process was known, the notion of robustness would vanish. But the notion of an *outlier* involves a large uncertainty about their origin and distribution. Consequently, using a wide normal distribution for p_o must be interpreted as a *back-up* model explaining observations which are highly

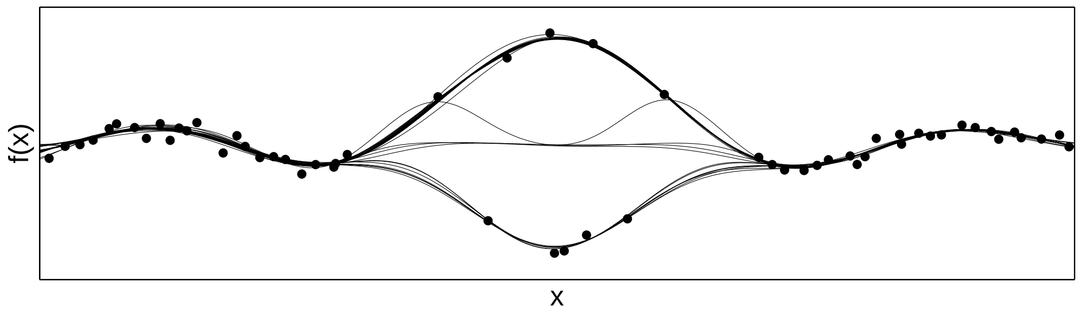


Figure 5.3.: Sampled functions from the posterior process of a GP regression model where the noise is modelled as a mixture of two normal distributions (5.3). The data (circles) have been designed in order to offer multiple alternative hypotheses to explain the data. Accordingly, the model shows uncertainty about whether the observations in the upper- and/or lower arc should be considered *outliers*. Therefore several hypotheses are mixed which leads to multimodality in the conflicting region. The samples have been generated using MCMC sampling as will be described in Section 5.3.2.

unlikely to stem from p_r.

We now turn towards the problem of Bayesian inference in the Gaussian process regression model with mixture noise of the form given by equation (5.3). According to eq. (3.4) the posterior distribution of latent function values $\mathbf{f}$ becomes

$$p(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) \;=\; \frac{\mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K})}{p(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})}\prod_{i=1}^{m}\left[(1-\pi)\,\mathcal{N}(y_i|f_i,\sigma_r^2)+\pi\,\mathcal{N}(y_i|f_i,\sigma_o^2)\right] \tag{5.4}$$

where the evidence is given by the integral

$$p(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi}) \;=\; \int \mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K})\prod_{i=1}^{m}\left[(1-\pi)\,\mathcal{N}(y_i|f_i,\sigma_r^2)+\pi\,\mathcal{N}(y_i|f_i,\sigma_o^2)\right]d\mathbf{f}\,. \tag{5.5}$$

In principle this integral is analytically solvable by rewriting it in terms of Gaussian integrals. However, this requires a change in the order of summation and the product, which would lead to a combinatorial explosion in the number of terms and the resulting posterior comes in the form of a mixture of 2^m normal distributions. Therefore, for problems of moderate size the huge number of components makes it computationally intractable and we have to resort to approximations.

Since the sampling distributions of errors (5.3) is not log-concave, the posterior (5.4) will in general be a multimodal distribution. Likewise, the posterior process for the mixture noise model is not a Gaussian process anymore but becomes a mixture of Gaussian processes which can have multi-modal marginal distributions $p(f_*|\mathcal{D},\mathbf{x}_*)$ as illustrated in Figure 5.3.

The multimodality can be understood intuitively by considering that each observation

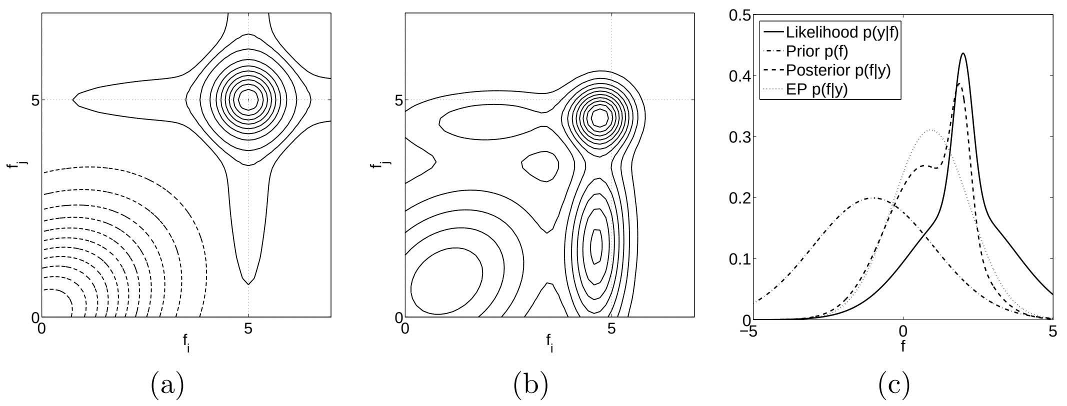


Figure 5.4.: Two dimensional illustration of the posterior for the GP regression model where the sampling distribution of errors is a mixture of normal distributions (5.3). Panel (a) shows contours of the prior (dashed), which is a bivariate normal distribution centred at zero, and the likelihood (solid) which is of the form (5.3) centred on the observation at $\mathbf{y} = [5, 5]^\top$. Panel (b) shows contours of the resulting posterior distribution which shows four modes of different magnitude. Each mode corresponds to an alternative assignment of the 2 observations to be outliers or regular observations. Panel (c) shows a one dimensional illustration and the EP approximation to the posterior.

can either be an outlier or a regular observation. Hence, for m observations there are 2^m different possible interpretations of the data, each representing an alternative assignment of the data into regular observations and outliers. Each such hypothesis can have a certain posterior probability which can give rise to a mode in the posterior distribution. An illustrative example is given in Figure 5.4.

The properties of the posterior make it difficult to approximate Bayesian inference. A Laplace approximation as described in Section 4.1 can only approximate one mode and practically it can not be guaranteed to find the mode of largest magnitude. Nevertheless, in experiments presented below, the Laplace approximation has shown to work relatively well, if only a few but extreme outliers are present. Note that in this case the posterior on $\mathbf{f}$ will in fact be very similar to a multivariate normal such that the assumptions of the Laplace's method are approximately met if the dominant mode is found. Furthermore, as will be shown in Section 5.3.1 Expectation Propagation can be implemented to approximate the posterior, although this comes with certain technical difficulties. Section 5.3.2 describes an MCMC implementation which exploits the (finite) scale-mixture representation.

5.3.1. Expectation Propagation Approximation

Expectation Propagation was described in its general form in Section 4.2. This section gives the necessary details for implementing EP for Gaussian process regression with mixture noise models. In general, let the sampling distribution of observations be a finite mixture

$$p(y|f(\mathbf{x}), \boldsymbol{\theta}) = \sum_{n=1}^{N} \pi_n \, p_n(y|f(\mathbf{x}), \boldsymbol{\theta}_n) \tag{5.6}$$

of N distributions with weights $\pi_n \geq 0$ such that $\pi_1 + \ldots + \pi_N = 1$. For implementing Expectation Propagation moments (4.22) of the form

$$m_k = \int f_i^k \, p(y|f_i, \boldsymbol{\theta}) \, \mathcal{N}(f_i|\mu_{\backslash i}, \sigma^2_{\backslash i}) \, df \tag{5.7}$$

have to be computed for $k = 0, 1, 2$ in order to update the site parameters using equations (4.25). The moment generating function of the mixture (5.6) can be written as

$$M(\lambda) = \sum_{n=1}^{N} \pi_n \int \exp(f_i \lambda) \, p_n(y|f_i, \boldsymbol{\theta}_n) \, \mathcal{N}(f_i|\mu_{\backslash i}, \sigma^2_{\backslash i}) \, df \tag{5.8}$$

such that the non-central moments of the mixture can be computed by combining the non-central moments of its components. Let m_k^n be the kth non-central moment of the nth component then

$$m_0 = \sum_{n=1}^{N} \pi_n m_0^n \tag{5.9a}$$

$$m_1 = \frac{1}{m_0} \sum_{n=1}^{N} \pi_n m_0^n m_1^n \tag{5.9b}$$

$$m_2 = \frac{1}{m_0} \sum_{n=1}^{N} \pi_n m_0^n m_2^n \tag{5.9c}$$

gives the moments of the mixture. The moments of the individual components can either be computed using numerical integration techniques, but if $p(y|f(\mathbf{x}), \boldsymbol{\theta})$ is, e.g., a normal or a Laplace distribution these can be computed analytically as will be described below.

For the normal distribution $p(y|f(\mathbf{x}), \boldsymbol{\theta}) = \mathcal{N}(y|f(\mathbf{x}), \sigma^2_{\mathfrak{n}})$ the non-central moments (5.7) are:

$$m_0 = \frac{1}{\sqrt{2\pi}\sqrt{\sigma^2_{\mathfrak{n}} + \sigma^2_{\backslash i}}} \exp\left(-\frac{(y - \mu_{\backslash i})^2}{2(\sigma^2_{\mathfrak{n}} + \sigma^2_{\backslash i})} \right) \tag{5.10a}$$

$$m_1 = \frac{\mu_{\backslash i}\sigma^2_{\mathfrak{n}} + y\sigma^2_{\backslash i}}{(\sigma^2_{\mathfrak{n}} + \sigma^2_{\backslash i})} \tag{5.10b}$$

$$m_2 = \frac{y^2\sigma^4_{\backslash i} + {\sigma^2_{\mathfrak{n}}}^2(\mu^2_{\backslash i} + \sigma^2_{\backslash i}) + \sigma^2_{\mathfrak{n}}(2y\mu_{\backslash i}\sigma^2_{\backslash i} + \sigma^4_{\backslash i})}{(\sigma^2_{\mathfrak{n}} + \sigma^2_{\backslash i})^2}\,. \tag{5.10c}$$

We now have everything necessary to implement EP for Gaussian process regres-

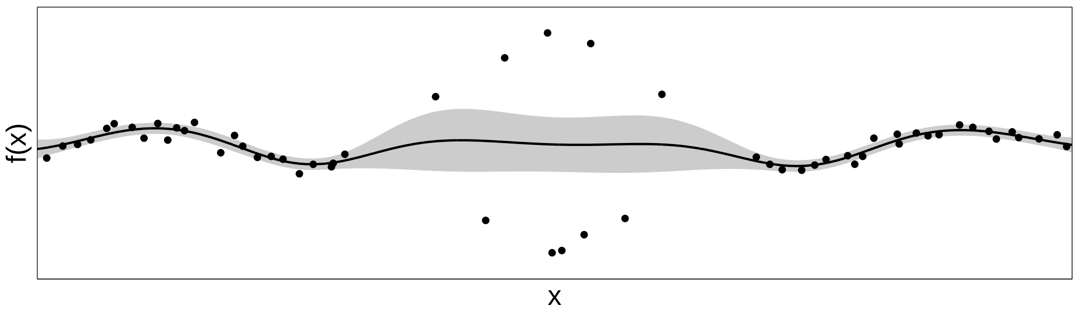


Figure 5.5.: The EP approximation for the Gaussian process regression model with normal mixture noise. The solid line describes the mean function of the approximate posterior process and the gray area covers ± 2 standard deviations. The data was introduced in Figure 5.3 to illustrate the multimodality of the posterior. In the EP approximation the posterior predictive distribution is unimodal and the mean function remains relatively constant over the region where alternative explanations of the data are possible. Note that a Gaussian process model with a single normal noise component interprets the data as pure noise without systematic structure, when the parameters are estimated using ML-II.

sion with normal mixture noise (5.3). The moments of the individual components are computed using eqs. (5.10) and combined using eqs. (5.9). See Figure 5.4(c) for a one dimensional illustration. Unfortunately, for the GP regression model with a mixture likelihood of two normal components, EP does not converge reliably for all values of the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. For every update of the site parameters (4.25) it has to be verified that the resulting covariance matrix $\mathbf{A}$ remains positive definite. In case an update gives rise to an invalid covariance matrix, the update has to be damped, e.g. by making the largest feasible update. The search for a feasible update of the parameters of a single site can involve several rank-one updates of the Cholesky decomposition of $\mathbf{A}$, which leads to a significant increase in computational costs, see also Kuss et al. (2005b).

5.3.2. Markov Chain Monte Carlo Sampling

An alternative way to approximate inference is Markov chain Monte Carlo sampling as described in Section 4.3. Sampling multimodal posterior distributions can be difficult if the modes are widely separated such that the chain is extremely unlikely to switch between the modes. In this situation the mixing behaviour of the chain has to be analysed carefully and it is advisable to run several chains with different initial states. Comparing the samples generated in independent simulations, it can be inspected whether the chains have mixed properly or whether chains stay in different modes.

The mixture of two normal distributions suggests an alternative sampling scheme by representing the noise variances $\sigma_1^2, \dots, \sigma_m^2$ of each likelihood term explicitly. For the mixture of two normal noise components we introduce binary variables $c_i \in \{0, 1\}$ which indicate whether an observation is attributed to the outlier component. Given the indicator variables the error on the ith observation is distributed according to

$$\varepsilon_i | c_i, \sigma_r^2, \sigma_o^2 \;\sim\; (1 - c_i)\,\mathcal{N}(\varepsilon_i | 0, \sigma_r^2) + c_i\,\mathcal{N}\,(\varepsilon_i | 0, \sigma_o^2) \tag{5.11}$$

where c_i indicates whether ε_i is attributed to the noise component with (larger) variance σ_o^2. This gives rise to the joint likelihood

$$p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta}) \;=\; \prod_{i=1}^{m} \left[(1 - c_i)\,\mathcal{N}(y_i | f_i, \sigma_r^2) + c_i \mathcal{N}\,(y_i | f_i, \sigma_o^2)\right] \;=\; \mathcal{N}(\mathbf{y}|\mathbf{f}, \boldsymbol{\Sigma}) \tag{5.12a}$$

where

$$\Sigma_{ii} \;=\; (1 - c_i)\sigma_r^2 + c_i \sigma_o^2 \tag{5.12b}$$

is a diagonal matrix containing the individual variances. Note that the likelihood parameters $\boldsymbol{\theta} = [\mathbf{c}, \sigma_r^2, \sigma_o^2]$ now also include the indicator variables.

We must specify prior distributions for the parameters of interest—namely for the parameters of the likelihood $p(\boldsymbol{\theta}|\boldsymbol{\xi})$ and for the parameters of the GP prior $p(\boldsymbol{\psi}|\boldsymbol{\xi})$, where we use $\boldsymbol{\xi}$ to collect parameters of both priors. The c_i are Bernoulli variables $p(c_i|\pi) = \text{Bernoulli}(\pi)$ where π is the fraction of samples attributed to noise variance σ_o^2. On π we put a beta prior $p(\pi|\alpha, \beta) = \text{Beta}(\alpha, \beta)$ introducing two more hyper-parameters. The choice of $p(\boldsymbol{\psi}|\boldsymbol{\xi})$ depends on the particular form of covariance function that is used. For example for positive parameters like length scales $\boldsymbol{\ell}$ and signal variances σ_s^2 using log-normal or gamma distributions are common choices. However, when comparing to other methods of approximate inference below, flat (constant, degenerate) priors will be used in order to minimise the influence of a particular choice.

The introduction of $\mathbf{c}$ is advantageous as it allows us to integrate over $\mathbf{f}|\mathbf{c}$ analytically such that the function values do not need to be represented explicitly. Instead of sampling $\mathbf{f}$ values we sample from

$$\begin{aligned} p(\boldsymbol{\theta}, \boldsymbol{\psi}|\mathcal{D}, \boldsymbol{\xi}) \;&\propto\; p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\psi})\; p(\boldsymbol{\theta}|\boldsymbol{\xi})\; p(\boldsymbol{\psi}|\boldsymbol{\xi}) && (5.13\text{a}) \\ &=\; \left[\int \mathcal{N}(\mathbf{y}|\mathbf{f}, \boldsymbol{\Sigma})\; \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})\, d\mathbf{f}\right] p(\boldsymbol{\theta}|\boldsymbol{\xi})\; p(\boldsymbol{\psi}|\boldsymbol{\xi}) && (5.13\text{b}) \\ &=\; \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K} + \boldsymbol{\Sigma})\, p(\boldsymbol{\theta}|\boldsymbol{\xi})\; p(\boldsymbol{\psi}|\boldsymbol{\xi}) && (5.13\text{c}) \end{aligned}$$

where $\mathbf{f}$ is integrated out using eq. (B.24) and $\boldsymbol{\theta}$ includes the indicator variables $\mathbf{c}$, compare to eq. (4.48).

To generate samples, a Markov chain has to be constructed whose state $[\boldsymbol{\theta}, \boldsymbol{\psi}]_t$ represents all the uncertain parameters we want to make inference about. The Markov chain is constructed according to the *Metropolis-Hastings* procedure and we employ different proposal techniques to exploit the structure of the model efficiently. The implemented sampling scheme iterates between *Gibbs* updates for the indicator variables $\mathbf{c}$ and π and *Hybrid* Monte Carlo updates for $\boldsymbol{\psi}$, σ_r^2, and σ_o^2.

First, however, we have to describe how the state is initialised. We initialise π by a sample from its prior distribution $\text{Beta}(\alpha, \beta)$ and consecutively sample $\mathbf{c}$ element-wise from a $\text{Bernoulli}(\pi)$. If proper prior distributions for $\boldsymbol{\psi}$, σ_r^2, and σ_o^2 are defined, initial values can be sampled from them. Alternatively we can use ML-II estimates for σ_r^2 and $\boldsymbol{\psi}$ from a model with simple normal noise. In this case, the initial value of σ_o^2 is simply set to $2\sigma_r^2$ afterwards. In the following we describe how we update the elements of the state—the value of the parameters—in the Markov chain.

Gibbs sampling as described in Section 4.3.2 is a common MCMC technique in which the state is updated component-wise by sampling from the respective conditional distributions of the components. The method is appealing since the proposed updates are always accepted and no further parameters are introduced. We can use this method to sample the fraction of outliers π and indicator variables $\mathbf{c}$. We have to sample from the conditional distribution of c_i given the values of all other variables $p(c_i|\mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\psi}, \mathcal{D}, \boldsymbol{\xi})$, where $\mathbf{c}^{\setminus i}$ denotes all elements of $\mathbf{c}$ except for the ith. We can decompose this probability

$$p(c_i|\mathbf{c}^{\setminus i}, \mathcal{D}, \sigma_r^2, \sigma_o^2, \boldsymbol{\psi}, \boldsymbol{\xi}) = \frac{p(\mathcal{D}|\mathbf{c}, \sigma_r^2, \sigma_o^2, \boldsymbol{\psi}, \boldsymbol{\xi})\, p(\mathbf{c}|\boldsymbol{\xi})}{p(\mathcal{D}|\sigma_r^2, \sigma_o^2, \boldsymbol{\psi}, \boldsymbol{\xi})\, p(\mathbf{c}^{\setminus i}|\mathcal{D}, \sigma_r^2, \sigma_o^2, \boldsymbol{\psi}, \boldsymbol{\xi})} \tag{5.14}$$

and observe that $p(c_i|\mathbf{c}^{\setminus i}, \mathcal{D}, \sigma_r^2, \sigma_o^2, \boldsymbol{\psi}, \boldsymbol{\xi}) \propto p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\xi})\, p(\mathbf{c}|\boldsymbol{\xi})$. Since c_i is a binary indicator it is Bernoulli distributed. The probability of success $\tilde{\pi}_i$ of this Bernoulli distribution can be computed by comparing $p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\xi})\, p(\mathbf{c}|\boldsymbol{\xi})$ evaluated for $c_i = 1$ and $c_i = 0$. Terms independent of c_i cancel and we find $\tilde{\pi}_i$ by looking at the ratio

$$\tilde{\pi}_i = \frac{\pi\, p(\mathcal{D}|c_i = 1, \mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\psi})}{(1-\pi)\, p(\mathcal{D}|c_i = 0, \mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\psi}) + \pi\, p(\mathcal{D}|c_i = 1, \mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\psi})} \tag{5.15}$$

which can be interpreted as the relative plausibility of the ith sample being an *outlier* given the current values of all other variables. Technically equation (5.15) compares the evidence evaluated for both values of c_i weighted by the current value of π.

The log evidence $\ln p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi})$ can be computed according to

$$\ln p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) = -\tfrac{m}{2}\ln(2\pi) - \tfrac{1}{2}\ln|\mathbf{K} + \boldsymbol{\Sigma}| - \tfrac{1}{2}\mathbf{y}^\top(\mathbf{K} + \boldsymbol{\Sigma})^{-1}\mathbf{y} \tag{5.16}$$

where the noise term $\boldsymbol{\Sigma}$, as given by eq. (5.12b), reflects the currently assumed noise on the observations (compare to (3.18c)). Hence, we can update c_i easily by a sample from Bernoulli$(\tilde{\pi}_i)$. Note that it would be computationally very costly to re-compute the log determinant and the inverse in eq. (5.16) for every proposal. Instead this step can be implemented using rank-one updates of the Cholesky decomposition of $\mathbf{K} + \boldsymbol{\Sigma}$.

A drawback of Gibbs sampling is that variables cannot change in a coordinated way. Take for example the data shown in Figure 5.3. In this case the posterior will have two modes of large magnitudes corresponding to whether the upper or lower points are considered outliers. A switch between the modes would require a coordinated change of all the c_i corresponding to the observations in the conflicting region. It is unclear how such a coordinated change of indicators could be proposed. However, we use *ordered overrelaxation* as described by Neal (2001) to improve the mixing behaviour.

The basic idea of ordered overrelaxation is to improve the mixing behaviour of Gibbs sampling schemes by encouraging larger changes in the variable based on order statistics of the conditional distribution. Here we only describe the special case of sampling from a Bernoulli distribution $c_i \sim \text{Bernoulli}(\tilde{\pi}_i)$. Let c_i^t denote the current value of c_i. Using ordered overrelaxation the distribution of the consecutive value c_i^{t+1} depends on value of c_i^t:

$$p(c_i^{t+1}|\tilde{\pi}_i, c_i^t = 0) \quad = \quad \text{Bernoulli}\left(\left\lceil \tfrac{\tilde{\pi}_i}{1-\tilde{\pi}_i} \right\rceil\right) \tag{5.17a}$$

$$p(c_i^{t+1}|\tilde{\pi}_i, c_i^t = 1) \quad = \quad \text{Bernoulli}\left(1 - \left\lceil \tfrac{1-\tilde{\pi}_i}{\tilde{\pi}_i} \right\rceil\right) \tag{5.17b}$$

where $\lceil \cdot \rceil$ denotes $\min(1, \cdot)$. The intuition is that the variable is encouraged to change its value as often as possible while leaving its distribution invariant, see Neal (2001) for details.

The next step in the sampling scheme is a Gibbs update of the mixing proportion π by a sample from $p(\pi|\mathbf{c}, \alpha, \beta) = \text{Beta}(\alpha + |\mathbf{c}|, \beta + m - |\mathbf{c}|)$ where $|\mathbf{c}|$ is the sum over elements of $\mathbf{c}$. This is a standard result, since the beta distribution is conjugate to the binomial, see for example O'Hagan (1994, ch. 1).

For updating the part of the state corresponding to $\boldsymbol{\psi}$, σ_r^2, and σ_o^2 we use Hybrid Monte Carlo where the potential energy is equivalent to eq. (5.16) plus the value of the respective log priors. Algorithm 1 provides a schematic overview of the sampling scheme.

Algorithm 1 MCMC sampling scheme for GP regression with mixture noise

```
Given: $\mathcal{D}, \alpha, \beta, \boldsymbol{\xi}$, and number and size of leapfrog steps for Hybrid MCMC
Initialisation:
Sample $\pi$ from Beta$(\alpha, \beta)$
Sample $\mathbf{c}$ element wise from Bernoulli$(\pi)$
Find initial values for $\boldsymbol{\psi}$, $\sigma_r^2$ and $\sigma_o^2$ (e.g. by maximising the evidence of a model with
simple Gaussian noise and setting $\sigma_o^2 \leftarrow 2\sigma_r^2$)
$t \leftarrow 0$
for each step of the Markov chain do
  $t \leftarrow t + 1$
  1. Gibbs updates (note that you can also do several sweeps over $\mathbf{c}$)
  for all $c_i$ do
    Compute $\tilde{\pi}_i$ by (5.15)
    Update $c_i$ by a sample from Bernoulli$(\tilde{\pi}_i)$ using ordered overrelaxation
  end for
  Sample $\pi$ from Beta$(\alpha + |\mathbf{c}|, \beta + N - |\mathbf{c}|)$
  2. Hybrid MC updates
  Update $\boldsymbol{\psi}$, $\sigma_r^2$, and $\sigma_o^2$ using Hybrid MCMC
  Save state $[\pi, \boldsymbol{\theta}, \boldsymbol{\psi}]_t$
end for
```

Having $t = 1, \ldots, T$ samples of the form $[\boldsymbol{\theta}, \boldsymbol{\psi}]_t$ the predictive distribution can be approximated by a mixture of T multivariate normal distributions:

$$\begin{aligned} p(\mathbf{f}_*|\mathcal{D}, \mathbf{X}_*, \boldsymbol{\xi}) &\approx \frac{1}{T}\sum_{t=1}^{T}\int p(\mathbf{f}_*|\mathbf{f}, \mathbf{X}, \mathbf{X}_*, \boldsymbol{\psi}_t)\, p(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}_t, \boldsymbol{\psi}_t)\, d\mathbf{f} \\ &= \frac{1}{T}\sum_{t=1}^{T}\mathcal{N}(\mathbf{f}_*|\mathbf{K}_*^\top(\mathbf{K}+\boldsymbol{\Sigma})^{-1}\mathbf{y}, \mathbf{K}_{**} - \mathbf{K}_*^\top(\mathbf{K}+\boldsymbol{\Sigma})^{-1}\mathbf{K}_*) \end{aligned} \tag{5.18}$$

where all covariance matrices ($\mathbf{K}$, $\mathbf{K}_*$, and $\mathbf{K}_{**}$) depend on $\boldsymbol{\psi}_t$ and the noise covariance matrix $\boldsymbol{\Sigma}$ depends on $\boldsymbol{\theta}_t$ (compare to (3.11)).

5.4. Regression with Student-t Noise

The most common approach to robust probabilistic regression is to assume that the sampling distribution of errors follows a Student-t distribution, because of its heavy tails. The probability density function of the Student-t distribution is given by

$$\mathcal{T}(x|\nu, \mu, \sigma) = \frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2)\sqrt{\nu\pi}\sigma}\left(1 + \frac{1}{\nu}\left(\frac{x-\mu}{\sigma}\right)^2\right)^{-(\nu+1)/2} \tag{5.19}$$

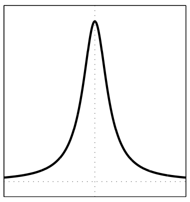


Figure 5.6.: P.d.f. of the Student-t distribution.

where ν is referred to as the degrees of freedom, μ is the location, and $\sigma > 0$ is a scale parameter. In the limit of $\nu \to \infty$ the Student-t distribution approaches the normal distribution while for $\nu = 1$ the Cauchy distribution is recovered as a particular case. The Student-t distribution is a scale-mixture (5.1) of infinitely many normal distributions

$$\mathcal{T}(x|2\alpha, \mu, \sqrt{\beta/\alpha}) = \int_0^\infty \mathcal{N}(x|\mu, \sigma^2)\, \mathrm{Inv}\Gamma(\sigma^2|\alpha, \beta)\, d\sigma^2\,, \tag{5.20}$$

where the scale-mixture distribution $p_M(\sigma_\mathrm{n}^2|\boldsymbol{\theta})$ is an inverse gamma distribution

$$\mathrm{Inv}\Gamma(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}x^{-(1+\alpha)}\exp\left(-\frac{\beta}{x}\right)\,, \tag{5.21}$$

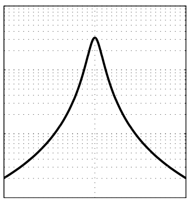


Figure 5.7.: Log of Student-t p.d.f.

with shape α and inverse scale β. Note that if σ^2 varies according to an inverse gamma distribution the precision σ^{-2} varies according to a gamma distribution.

The probability density function of the Student-t distribution is not log-concave. Therefore the posterior $p(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi})$ in the Gaussian process regression model with Student-t noise can be multimodal. An EP approximation could be implemented but the necessary moments must be approximated using numerical integration techniques. Furthermore, it could be expected that the EP approximation would show the same problems as for the mixture models described above, due to the non-concavity of the log likelihood. Nevertheless, a variational approximation can be made, as will be described in the following section. Section 5.4.2 provides a description of a Markov chain

Monte Carlo sampling scheme, again exploiting the scale-mixture representation.

5.4.1. A Variational Approximation

Tipping and Lawrence (2003, 2005) describe a variational approximation for regression with Student-t noise in the Relevance Vector Machine framework (Tipping, 2001). A similar approximation can also be applied to find a Gaussian approximation

$$p(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) \;\approx\; q(\mathbf{f}|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) \;=\; \mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A}) \tag{5.22}$$

to the posterior in the Gaussian process regression model with Student-t noise. The approach is based on the property that the Student-t distribution can be written as a scale-mixture (5.20) by representing the individual variances $\boldsymbol{\sigma}^2 = [\sigma_1^2,\ldots,\sigma_m^2]^\top$ explicitly. The distribution of $\mathbf{y}$ given $\mathbf{f}$ and the individual variances is

$$p(\mathbf{y}|\mathbf{f},\boldsymbol{\sigma}^2) \;=\; \prod_{i=1}^{m}\mathcal{N}(y_i|f_i,\sigma_i^2) \;=\; \mathcal{N}(\mathbf{y}|\mathbf{f},\boldsymbol{\Sigma}) \tag{5.23}$$

where $\boldsymbol{\Sigma} = \operatorname{diag}(\boldsymbol{\sigma}^2)$ and using a joint inverse gamma prior on the variances

$$p_M(\boldsymbol{\sigma}^2|\boldsymbol{\theta}) \;=\; \prod_{i=1}^{m}\operatorname{Inv}\Gamma(\sigma_i^2|\alpha,\beta) \tag{5.24}$$

the Student-t model is obtained. The joint posterior of latent function values and variances is

$$p(\mathbf{f},\boldsymbol{\sigma}^2|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) \;=\; \frac{p(\mathbf{y}|\mathbf{f},\boldsymbol{\sigma}^2)\,p(\boldsymbol{\sigma}^2|\boldsymbol{\theta})\,\mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K})}{p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta},\boldsymbol{\psi})} \tag{5.25}$$

which cannot be computed in closed form since the integral in the denominator, i.e. the evidence

$$p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta},\boldsymbol{\psi}) \;=\; \int p(\mathbf{y}|\mathbf{f},\boldsymbol{\sigma}^2)\,p(\boldsymbol{\sigma}^2|\boldsymbol{\theta})\,\mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K})\,d\mathbf{f}\,d\boldsymbol{\sigma}^2\,, \tag{5.26}$$

is analytically intractable. The idea of the variational approximation is to minimise the Kullback-Leibler divergence $\mathrm{KL}(q\,||\,p)$ (as defined in Appendix B.2.3.6) where q is the approximation and p the true posterior distribution (5.25), see for instance Beal (2003). Simultaneously we are interested in finding ML-II estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. Combining these two desiderata leads to an expectation maximisation (EM) type algorithm as proposed by Dempster et al. (1977).

The approximation q to the posterior is chosen to be of a factorising form

$$p(\mathbf{f},\boldsymbol{\sigma}^2|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi}) \;\approx\; q(\mathbf{f})q(\boldsymbol{\sigma}^2) \tag{5.27a}$$

where the approximate posterior distributions of $\mathbf{f}$ and $\boldsymbol{\sigma}^2$:

$$\begin{aligned} q(\mathbf{f}) &= \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{A}) && (5.27b) \\ q(\boldsymbol{\sigma}^2) &= \prod_{i=1}^{m} \mathrm{Inv}\Gamma(\sigma_i^2|\tilde{\alpha}_i, \tilde{\beta}_i) && (5.27c) \end{aligned}$$

are of the same parametric family as their respective prior distributions.

The variational approximation is found by maximising the evidence (5.26) using the EM algorithm while $\mathbf{f}$ and $\boldsymbol{\sigma}^2$ are treated as latent variables. The evidence cannot be evaluated in closed form, but Jensen's inequality can be used to obtain a lower bound $\mathcal{V}$:

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\psi}) = \int p(\mathbf{y}, \boldsymbol{\sigma}^2, \mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\psi})\, d\mathbf{f}\, d\boldsymbol{\sigma}^2 & \\ \geq \mathcal{V}(q(\mathbf{f}), q(\boldsymbol{\sigma}^2), \boldsymbol{\theta}, \boldsymbol{\psi}) &= \int q(\mathbf{f})\, q(\boldsymbol{\sigma}^2) \ln \frac{p(\mathbf{y}, \boldsymbol{\sigma}^2, \mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\psi})}{q(\mathbf{f})\, q(\boldsymbol{\sigma}^2)}\, d\mathbf{f}\, d\boldsymbol{\sigma}^2 \quad (5.28) \end{aligned}$$

which can be computed explicitly. Expanding the variational lower bound leads to a sum of five expectations:

$$\mathcal{V} = \mathop{\mathbb{E}}_{q(\mathbf{f})q(\boldsymbol{\sigma}^2)} \ln p(\mathbf{y}|\mathbf{f}, \boldsymbol{\sigma}^2) + \mathop{\mathbb{E}}_{q(\boldsymbol{\sigma}^2)} \ln p(\boldsymbol{\sigma}^2|\boldsymbol{\theta}) + \mathop{\mathbb{E}}_{q(\mathbf{f})} \ln \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}) + \mathcal{H}(q(\mathbf{f})) + \mathcal{H}(q(\boldsymbol{\sigma}^2)) \quad (5.29)$$

where $\mathcal{H}$ denotes the entropy of a distribution, see Appendix B.2.3.5. The EM algorithm iterates between updates of the approximate distributions and the parameter values. In the so called E-step, the *expectation* step, $\mathcal{V}$ is maximised in $q(\mathbf{f})$ and $q(\boldsymbol{\sigma}^2)$ for fixed values of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. Each E-step is followed by an M-step, a *maximisation* step, in which new values of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ are found by maximising $\mathcal{V}$ for fixed $q(\mathbf{f})$ and $q(\boldsymbol{\sigma}^2)$. The algorithm iterates E-steps and M-steps until convergence of $\mathcal{V}$ which is guaranteed not to decrease in each iteration. In the following sections the E-step and the M-step will be described in detail and Algorithm 2 (on page 75) will summarise the computations.

5.4.1.1. The E-step

In the E-step the q distributions are estimated as to maximise $\mathcal{V}$ for given $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ which is equivalent to minimising the Kullback-Leibler divergence between the approximation and the posterior distribution:

$$\mathop{\mathrm{argmax}}_{q(\mathbf{f}), q(\boldsymbol{\sigma}^2)} \mathcal{V}(q(\mathbf{f}), q(\boldsymbol{\sigma}^2), \boldsymbol{\theta}, \boldsymbol{\psi}) = \mathop{\mathrm{argmin}}_{q(\mathbf{f}), q(\boldsymbol{\sigma}^2)} \mathrm{KL}(q(\mathbf{f})q(\boldsymbol{\sigma}^2)||p(\mathbf{f}, \boldsymbol{\sigma}^2|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi})) \quad (5.30)$$

The true posterior would be the minimiser of this KL divergence but since the solution is constrained to be of the form $q(\mathbf{f})q(\boldsymbol{\sigma}^2)$ it is unlikely that this minimum will be attained. Instead a sequential approach will be taken in which one of the distributions is fixed while optimising over the respective other distribution

$$\begin{aligned} q(\mathbf{f}) &= \operatorname*{argmin}_{q(\mathbf{f})} \mathrm{KL}(q(\mathbf{f})\, q(\boldsymbol{\sigma}^2) \,||\, p(\mathbf{f}, \boldsymbol{\sigma}^2 | \mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi})) && (5.31a) \\ q(\boldsymbol{\sigma}) &= \operatorname*{argmin}_{q(\boldsymbol{\sigma})} \mathrm{KL}(q(\mathbf{f})\, q(\boldsymbol{\sigma}^2) \,||\, p(\mathbf{f}, \boldsymbol{\sigma}^2 | \mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi})) && (5.31b) \end{aligned}$$

subject to the constraint that the distributions are of their respective form given by eqs. (5.27). Expanding equations (5.31) and neglecting terms that do not depend on $q(\mathbf{f})$ and $q(\boldsymbol{\sigma}^2)$ respectively, the above problems can be shown to be equivalent to

$$\begin{aligned} q(\mathbf{f}) &= \operatorname*{argmin}_{q(\mathbf{f})} \mathop{\mathbb{E}}_{q(\boldsymbol{\sigma}^2)} \mathrm{KL}(q(\mathbf{f}) \,||\, p(\mathbf{f} | \mathcal{D}, \boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{\psi})) && (5.32a) \\ q(\boldsymbol{\sigma}^2) &= \operatorname*{argmin}_{q(\boldsymbol{\sigma}^2)} \mathop{\mathbb{E}}_{q(\mathbf{f})} \mathrm{KL}(q(\boldsymbol{\sigma}^2) \,||\, p(\boldsymbol{\sigma}^2 | \mathcal{D}, \mathbf{f}, \boldsymbol{\theta}, \boldsymbol{\psi}))\ . && (5.32b) \end{aligned}$$

Using the property that both $p(\mathbf{f})$ and $q(\boldsymbol{\sigma}^2)$ are in the exponential family and that the inverse gamma is conjugate to the normal distribution the following update equations can be derived

$$\begin{aligned} q(\mathbf{f}) &\propto \exp \mathop{\mathbb{E}}_{q(\boldsymbol{\sigma}^2)} \ln p(\mathbf{f} | \mathcal{D}, \boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{\psi}) && (5.33a) \\ q(\boldsymbol{\sigma}^2) &\propto \exp \mathop{\mathbb{E}}_{q(\mathbf{f})} \ln p(\boldsymbol{\sigma}^2 | \mathcal{D}, \mathbf{f}, \boldsymbol{\theta}, \boldsymbol{\psi})\ . && (5.33b) \end{aligned}$$

For the mean and covariance of $q(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{A})$ one obtains

$$\begin{aligned} \mathbf{A} &= (\mathbf{K}^{-1} + \boldsymbol{\Sigma}^{-1})^{-1} && (5.34a) \\ \mathbf{m} &= \mathbf{A}\boldsymbol{\Sigma}^{-1}\mathbf{y} && (5.34b) \end{aligned}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix such that $\Sigma_{ii}^{-1} = \mathbb{E}[\sigma_i^{-2}] = \tilde{\alpha}_i / \tilde{\beta}_i$. The update equation of $\mathbf{A}$ can be rewritten as

$$\mathbf{A} = \mathbf{K} - \mathbf{K}\boldsymbol{\Sigma}^{-1/2}(\mathbf{I} + \boldsymbol{\Sigma}^{-1/2}\mathbf{K}\boldsymbol{\Sigma}^{-1/2})^{-1}\boldsymbol{\Sigma}^{-1/2}\mathbf{K} \qquad (5.35)$$

which is numerically more stable since the eigenvalues of the inverse are known to be larger than one and $\boldsymbol{\Sigma}^{-1/2}$ is a positive definite diagonal matrix. Note that the equations (5.34) can also be derived directly by computing the derivatives of the variational lower bound (5.29) with respect to $\mathbf{A}$ and $\mathbf{m}$ and equating them with zero.

Analogously, from eq. (5.33b) the update equations of the parameters of $q(\boldsymbol{\sigma}^2)$ come in the form of

$$\begin{aligned} \tilde{\alpha}_i &= \alpha + \tfrac{1}{2} && (5.36a) \\ \tilde{\beta}_i &= \beta + \tfrac{1}{2}\left((y_i - m_i)^2 + A_{ii}\right) && (5.36b) \end{aligned}$$

from which we can see that all elements of $\tilde{\boldsymbol{\alpha}}$ have the same value while $\tilde{\boldsymbol{\beta}}$ depends on both $\mathbf{m}$ and $\mathbf{A}$.

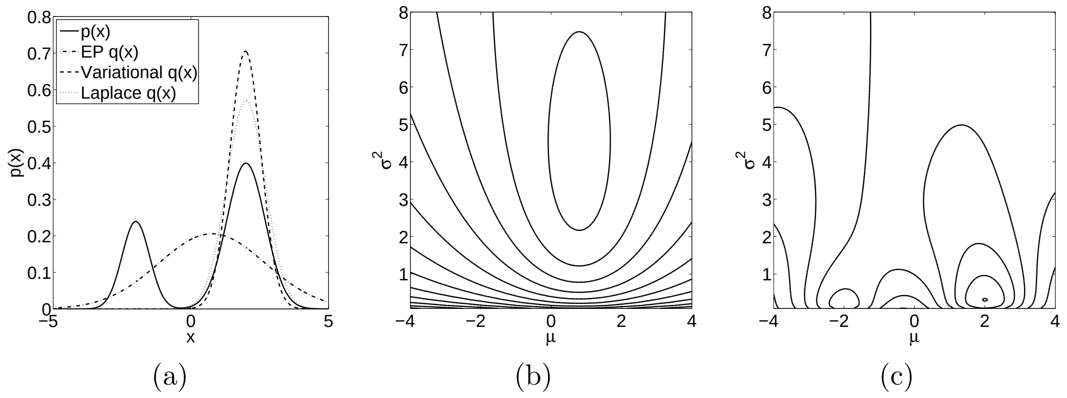


Figure 5.8.: Illustration of the difference between the approximations minimising $\mathrm{KL}(p\,||\,q)$ and $\mathrm{KL}(q\,||\,p)$. Panel (a) shows a bimodal distribution $p(x)$ and approximations thereof $q(x) = \mathcal{N}(x|\mu, \sigma^2)$. The distribution $p(x)$ is a weighted combinations of two normal distributions $\mathcal{N}(x|-2, 0.5^2)$ and $\mathcal{N}(x|2, 0.7^2)$. Panel (b) shows contours of $\mathrm{KL}(p\,||\,q)$ as a function of μ and σ^2. The approximation EP $q(x)$ corresponds to the minimum of $\mathrm{KL}(p\,||\,q)$. Panel (c) shows contours of $\mathrm{KL}(q\,||\,p)$ as minimised in variational approximations. Note that the function has two local minima corresponding to the modes of $p(x)$. The approximation Variational $q(x)$ in Panel (a) corresponds to the local minimum close to $\mu = 2$ and is very similar to a Laplace approximation of this mode.

The update equations (5.34) and (5.36) form a set of coupled nonlinear equations. In each E-step one iterates update equations (5.34) and (5.36) until convergence. The iterations are known to converge since each update does not decrease $\mathcal{V}$. In the simulations we have done usually three to five iterations were sufficient.

Note that from a conceptual point of view the variational approximation attempts to minimise $\mathrm{KL}(q\,||\,p)$ in each E-step, where q is the approximation and p the true posterior. In contrast, Expectation Propagation can be interpreted as an approximate minimisation of $\mathrm{KL}(p\,||\,q)$ (Minka, 2001a). Since the Kullback-Leibler divergence (B.30) is asymmetric, different properties of the solutions can be expected, see Figure 5.8 for an illustration. Intuitively the EP approximation attempts to be nonzero wherever the true posterior is nonzero, which corresponds to a coverage of the whole posterior by the approximation. By minimising the KL divergence the other way around, the variational approximation will not necessarily cover the whole posterior but can focus on one mode, as can be seen in the examples given in Figure 5.9.

5.4.1.2. The M-step

In the M-Step the variational lower bound $\mathcal{V}$ is maximised with respect to the likelihood parameters $\boldsymbol{\theta} = [\alpha, \beta]^\top$ and the hyper-parameters of the Gaussian process prior $\boldsymbol{\psi}$, for example using a conjugate gradient optimisation routine. Therefore the value of $\mathcal{V}$ as given by equation (5.29) has to be computed.

To compute the individual terms of $\mathcal{V}$ as given by eq. (5.29) several integrals involving the inverse gamma distribution have to be evaluated. Note that if x is distributed $\mathrm{Inv}\Gamma(x|\alpha,\beta)$, then $\mathbb{E}[x] = \beta/(\alpha+1)$, $\mathbb{E}[x^{-1}] = \alpha/\beta$, and $\mathbb{E}[\ln x] = \ln\beta - \psi(\alpha)$, where $\psi(x) = \partial \ln\Gamma(x)/\partial x$ is the digamma function (Abramowitz and Stegun, 1965, ch. 6.3). The integral over a product of an inverse gamma distribution and the log of another inverse gamma distribution is

$$\int \mathrm{Inv}\Gamma(x|\tilde{\alpha},\tilde{\beta}) \ln \mathrm{Inv}\Gamma(x|\alpha,\beta)\,dx \;=\; -\frac{\tilde{\alpha}}{\tilde{\beta}}\beta - (1+\alpha)\ln\tilde{\beta} + \alpha\ln\beta - \ln\Gamma(\alpha) + (1+\alpha)\psi(\tilde{\alpha}) \quad (5.37)$$

from which the entropy of the inverse gamma distribution

$$\mathcal{H}(\mathrm{Inv}\Gamma(x|\alpha,\beta)) \;=\; \alpha + \ln\beta + \ln\Gamma(\alpha) - (1+\alpha)\psi(\alpha) \quad (5.38)$$

can be derived as a special case.

Using these integrals and the results for multivariate normal distributions given in Appendix B.2.3 the first term of the variational lower bound (5.29) can be computed in which the expectation of the multivariate normal term $p(\mathbf{y}|\mathbf{f},\boldsymbol{\sigma}^2)$ is taken with respect to both q distributions:

$$\underset{q(\mathbf{f})q(\boldsymbol{\sigma}^2)}{\mathbb{E}} \ln p(\mathbf{y}|\mathbf{f},\boldsymbol{\sigma}^2) = \int q(\boldsymbol{\sigma}^2)\left[\int \mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A})\ln\mathcal{N}(\mathbf{y}|\mathbf{f},\mathrm{diag}(\boldsymbol{\sigma}^2))d\mathbf{f}\right]d\boldsymbol{\sigma}^2 = -\tfrac{m\ln(2\pi)}{2} - \tfrac{1}{2}\sum_{i=1}^{m}(\ln\tilde{\beta}_i - \psi(\tilde{\alpha}_i)) - \tfrac{1}{2}(\mathbf{y}-\mathbf{m})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{m}) - \tfrac{1}{2}\,\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{A})\,. \quad (5.39)$$

Note that the resulting term does not depend on either $\boldsymbol{\theta}$ or $\boldsymbol{\psi}$, hence it is unnecessary to compute it in the M-step. The second term of eq. (5.29) is an integral over a product of an inverse gamma distributions and the log of another inverse gamma distribution. Using eq. (5.37) this leads to

$$\underset{q(\boldsymbol{\sigma}^2)}{\mathbb{E}} \ln p(\boldsymbol{\sigma}^2|\boldsymbol{\theta}) \;=\; \sum_{i=1}^{m}\left[-\frac{\tilde{\alpha}_i\beta}{\tilde{\beta}_i} + \alpha\ln\beta - \ln\Gamma(\alpha) + (1+\alpha)(\psi(\tilde{\alpha}_i) - \ln\tilde{\beta}_i)\right] \quad (5.40)$$

which depends on $\boldsymbol{\theta}$. The third term of the variational lower bound (5.29) is an integral over the product of a multivariate normal and the log of another multivariate normal distribution

$$\underset{q(\mathbf{f})}{\mathbb{E}} \ln\mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K}) = -\tfrac{m\ln(2\pi)}{2} - \tfrac{1}{2}\ln|\mathbf{K}| - \tfrac{1}{2}\mathbf{m}^\top\mathbf{K}^{-1}\mathbf{m} - \tfrac{1}{2}\,\mathrm{tr}(\mathbf{K}^{-1}\mathbf{A}) \quad (5.41)$$

which depends on $\boldsymbol{\psi}$. The remaining two terms are the entropy of the multivariate

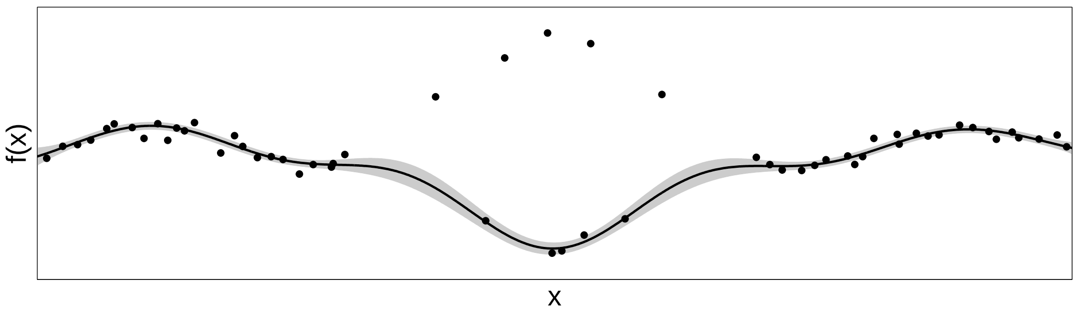


Figure 5.9.: The variational approximation for the Gaussian process regression model with Student-t noise. The solid line describes the mean function of the approximate posterior process and the gray area covers ± 2 standard deviations. The data was introduced in Figure 5.3 to illustrate the multimodality of the posterior, see also Figure 5.5. In the shown example the approximation considers the upper points to be outliers, which means it only approximates one mode of the posterior. For different initialisations of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ also the opposite solution occurs in which only the lower points in the central region are considered outliers. This behaviour can be explained by the discussion in the context of Figure 5.8. Note that in this respect the variational approximation seems to be similar to a Laplace approximation.

normal $q(\mathbf{f})$ and the inverse gamma distributions $q(\boldsymbol{\sigma})$

$$\mathcal{H}(q(\mathbf{f})) = \tfrac{m}{2}\ln(2\pi) + \tfrac{1}{2}\ln|\mathbf{A}| + \tfrac{m}{2} \tag{5.42}$$

$$\mathcal{H}(q(\boldsymbol{\sigma}^2)) = \sum_{i=1}^{m}\Big(\tilde{\alpha}_i + \ln\tilde{\beta}_i + \ln\Gamma(\tilde{\alpha}_i) - (1+\tilde{\alpha}_i)\psi(\tilde{\alpha}_i)\Big) \tag{5.43}$$

which can be computed using eq. (5.38) and eq. (B.29) and which are independent of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$.

From the equations above follows that maximising $\mathcal{V}$ in the M-Step is equivalent to maximising the sum of eq. (5.40) and eq. (5.41). See Algorithm 2 for a summary of the variational approximation scheme. Note that the optimisation problems in the E-Step and the M-Step are non-convex, so the scheme is prone to local maxima problems. In practice, it is therefore advisable to run the scheme several times using different initial values and to select the approximation which shows the highest overall value of $\mathcal{V}$. Figure 5.9 shows the resulting variational approximation for the Gaussian process regression model with Student-t noise for ML-II estimates of the parameters.

5.4.1.3. Prediction

After convergence of the EM algorithm, i.e. convergence of $\mathcal{V}$, we have the approximation $p(\mathbf{f}|\mathcal{D},\boldsymbol{\psi},\boldsymbol{\theta}) \approx q(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A})$ which implies an approximation to the posterior process. By substituting eqs. (5.34) into eqs. (4.5) the mean and covariance function of

Algorithm 2 Variational Approximation for GP Regression with Student-t noise

```
Initialisation: $\tilde{\boldsymbol{\alpha}} \leftarrow \mathbf{1}$, $\tilde{\boldsymbol{\beta}} \leftarrow \mathbf{1}$, set $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ to random (positive) values
Compute $\mathbf{K}$ from $\mathbf{X}$ and $\boldsymbol{\psi}$
repeat
  E-step:
  repeat
    $\boldsymbol{\Sigma}^{-1} \leftarrow \operatorname{diag}(\tilde{\boldsymbol{\alpha}} \oslash \tilde{\boldsymbol{\beta}})$        [Let $\oslash$ denote element-wise division]
    $\mathbf{A} \leftarrow \mathbf{K} - \mathbf{K}\boldsymbol{\Sigma}^{-1/2}(\mathbf{I} + \boldsymbol{\Sigma}^{-1/2}\mathbf{K}\boldsymbol{\Sigma}^{-1/2})^{-1}\boldsymbol{\Sigma}^{-1/2}\mathbf{K}$        [Numerically stable]
    $\mathbf{m} \leftarrow \mathbf{A}\boldsymbol{\Sigma}^{-1}\mathbf{y}$
    $\tilde{\boldsymbol{\alpha}} \leftarrow \mathbf{1}(\alpha + \frac{1}{2})$
    $\tilde{\boldsymbol{\beta}} \leftarrow \beta + \frac{1}{2}(\operatorname{diag}(\mathbf{A}) + (\mathbf{y} - \mathbf{m})^2)$
  until convergence
  M-step:
  $[\boldsymbol{\theta}, \boldsymbol{\psi}] \leftarrow \operatorname{argmax} \mathcal{V}(\boldsymbol{\theta}, \boldsymbol{\psi})$        [Note that only (5.40) and (5.41) depend on $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$]
  Compute $\mathbf{K}$ from $\mathbf{X}$ and $\boldsymbol{\psi}$
until convergence of $\mathcal{V}$
```

the Gaussian process approximation to the posterior process can be computed:

$$m_*(\mathbf{x}) = \mathbf{k}(\mathbf{x})^\top(\mathbf{K} + \boldsymbol{\Sigma})^{-1}\mathbf{y} \tag{5.44a}$$

$$k_*(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x})^\top(\mathbf{K} + \boldsymbol{\Sigma})^{-1}\mathbf{k}(\mathbf{x}')\,. \tag{5.44b}$$

The predictive distribution of noisy targets y_* is given by

$$p(y_*|\mathbf{x}_*, \alpha, \beta) = \int \mathcal{T}(y_*|2\alpha, f_*, \sqrt{\beta/\alpha})\,\mathcal{N}(f_*|\mu_*, \sigma_*^2)\,df_* \tag{5.45}$$

which cannot be computed analytically but Gauss-Hermite quadrature can be used to solve this one-dimensional integral for a given y_* efficiently.

5.4.2. Markov Chain Monte Carlo Sampling

Several alternative Markov chain Monte Carlo schemes can be used to implement approximate inference in the Gaussian process regression model with Student-t noise. As described in Section 4.3.4 the generic approach is to represent the function values $\mathbf{f}$ explicitly and to sample from

$$p(\mathbf{f}, \boldsymbol{\theta}, \boldsymbol{\psi}|\mathcal{D}, \boldsymbol{\xi}) \propto p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})\,p(\mathbf{f}|\mathbf{X}, \boldsymbol{\psi})\,p(\boldsymbol{\theta}|\boldsymbol{\xi})\,p(\boldsymbol{\psi}|\boldsymbol{\xi}) \tag{5.46}$$

using Hybrid Monte Carlo. The disadvantage of this approach is that the elements of $\mathbf{f}$ can be highly correlated such that the mixing behaviour can be poor. Particularly if the posterior is multimodal, as for the Student-t model, switching between modes can be difficult.

If the likelihood can be represented as a scale-mixture of normal distributions, an alternative implementation usually shows to be easier to sample from. The idea is to represent the noise variances $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\sigma}^2)$ explicitly and to integrate over $\mathbf{f}|\boldsymbol{\sigma}^2$ analytically. Hybrid Monte Carlo is used to sample from

$$
\begin{aligned}
p(\boldsymbol{\sigma}^2, \boldsymbol{\theta}, \psi|\mathcal{D}, \boldsymbol{\xi}) &\propto \left[\int \mathcal{N}(\mathbf{y}|\mathbf{f}, \boldsymbol{\Sigma})\,\mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})d\mathbf{f}\right] p_M(\boldsymbol{\sigma}^2|\boldsymbol{\theta})\,p(\boldsymbol{\theta}|\boldsymbol{\xi})p(\psi|\boldsymbol{\xi}) \quad (5.47\text{a})\\
&\propto \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}+\boldsymbol{\Sigma})\,p_M(\boldsymbol{\sigma}^2|\boldsymbol{\theta})\,p(\boldsymbol{\theta}|\boldsymbol{\xi})\,p(\psi|\boldsymbol{\xi}) \quad (5.47\text{b})
\end{aligned}
$$

where $p_M(\boldsymbol{\sigma}^2|\boldsymbol{\theta})$ is the scale-mixture distribution, which for Student-t noise is a product of inverse gamma distributions as given by eq. (5.24). Note that the elements of $\boldsymbol{\sigma}^2$ are constraint to be positive. This is implemented by sampling $\ln \boldsymbol{\sigma}^2$, which requires an additional Jacobian term in order to correct for the log transformation.

After having generated samples of the posterior (5.47) the predictive distribution $p(\mathbf{f}_*|\mathcal{D}, \mathbf{X}_*, \boldsymbol{\xi})$ can be approximated using eq. (5.18). The MCMC scheme described for the mixture model in Section 5.3.2 can be seen as a special case in which $p_M(\boldsymbol{\sigma}^2|\boldsymbol{\theta})$ constrains the elements of $\boldsymbol{\sigma}^2$ to take one of to values, i.e. σ_r^2 or σ_o^2. Because of the discrete nature of the indicator variables $\mathbf{c}$, Gibbs sampling had to be used instead of Hybrid MCMC as used here for the continuous $\boldsymbol{\sigma}^2$.

5.5. Regression with Laplace Noise

The practical difficulties in approximating Bayesian inference for the models discussed in previous sections were caused by the multimodality of the posterior. As described, in the context of robust models multimodal posteriors are meaningful in the sense that several alternative explanations of the data coexist under the posterior.

However, in order to assure unimodality of the posterior in Gaussian process models the sampling distribution of errors—the likelihood—has to be log-concave, which effectively limits the thickness of its tails. Therefore, Laplace noise is of particular interest because its density function falls off maximally slowly in the tails while still being log-concave. The Laplace distribution, which is also known as the double exponential distribution, has the probability density function

$$
\mathrm{Laplace}(x|\mu, s) = \frac{1}{2s}\exp\left(-\frac{|x-\mu|}{s}\right) \quad (5.48)
$$

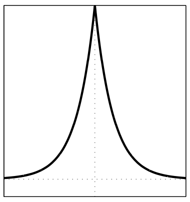


Figure 5.10.: The p.d.f. of the Laplace distribution.

which is log-concave, but its derivative with respect to x is not continuous at zero. The Laplace distribution is a scale-mixture of normal distributions (5.1) where the scale-mixture $p_M(\sigma^2|\boldsymbol{\theta})$ is an exponential distribution. Let $\mathrm{Exponential}(x|\beta) = \beta\exp(-\beta x)$ denote the exponential distribution where $x \geq 0$ and $\mathbb{E}[x] = 1/\beta$. The Laplace distribution

$$
\mathrm{Laplace}(x|\mu, s) = \int \mathcal{N}(x|\mu, \sigma^2)\,\mathrm{Exponential}(\sigma^2|\beta)\,d\sigma^2 \quad (5.49)
$$

is obtained as a scale-mixture of normal distributions and $s = 1/\sqrt{2\beta}$.

Using Laplace noise in a Gaussian process regression model can be seen as a compro-

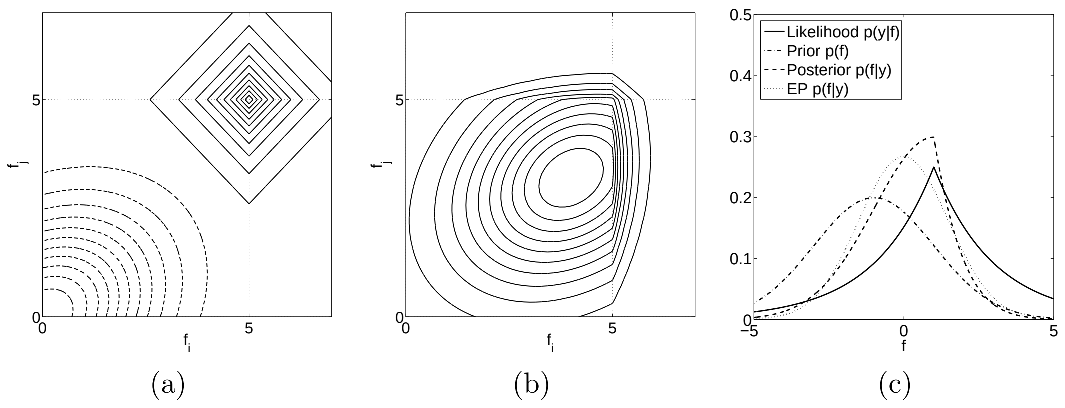


Figure 5.11.: Two dimensional illustration of the posterior for the GP regression model with Laplace noise. Panel (a) shows contours of the prior (dashed), which is a bivariate normal distribution centred at zero, and the likelihood centred on the observation at $\mathbf{y} = [5, 5]^\top$. Panel (b) shows contours of the resulting posterior distribution which is unimodal. Panel (c) shows a one dimensional illustration of the posterior and its EP approximation. Note that at the maximum of the posterior the first derivative has a discontinuity such that a Laplace approximation is not properly defined. Compare to Figure 5.4.

mise between the desideratum of outlier resistance and practical computational advantages that come with the unimodality of the posterior. Assuming i.i.d. Laplace noise, the likelihood of f comes in the form of

$$p(\mathbf{y}|\mathbf{f}, s) \;=\; \prod_{i=1}^{m} \text{Laplace}(y_i|f(\mathbf{x}_i), s)\,. \tag{5.50}$$

The assumption of Laplace noise is rarely found in the literature on Bayesian regression models. However, in frequentist robust regression maximum likelihood estimates of the form

$$f^\star \;=\; \operatorname*{argmax}_{f\in\mathcal{F}} p(\mathbf{y}|\mathbf{f}, s) \;=\; \operatorname*{argmin}_{f\in\mathcal{F}} \sum_{i=1}^{m} |y_i - f(\mathbf{x}_i)| \tag{5.51}$$

are common where $\mathcal{F}$ is usually chosen to be the class of linear functions (Rousseeuw and Leroy, 1987). The method appears under various names including: "least absolute deviation", "least absolute errors", "least absolute value", and L_1 regression. Although being less sensitive with respect to outliers, the breakdown point of L_1 regression is still 0%, i.e. a single observation can have an arbitrary large effect on the maximum likelihood estimate. See Narula and Wellington (1982) for a review of frequentist linear L_1 regression.

The log likelihood of the Laplace noise model can also be related to the ϵ-insensitive loss function

$$|y - f(\mathbf{x})|_\epsilon \;=\; \max\{0, |y - f(\mathbf{x})| - \epsilon\} \tag{5.52}$$

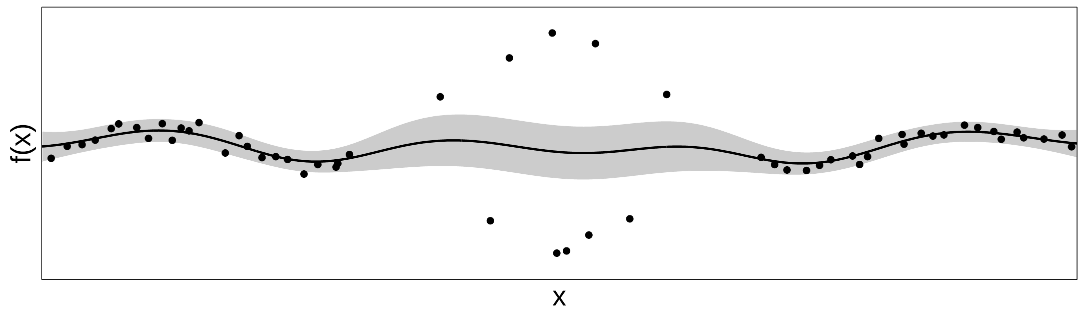


Figure 5.12.: The EP approximation for the Gaussian process regression model with Laplace noise. The solid line describes the mean function of the approximate posterior process and the gray area covers ± 2 standard deviations. The data was introduced in Figure 5.3 to illustrate the multimodality of the posterior, see also Figures 5.5 and 5.9. The observations in the central region are considered outliers, i.e. these observations show large deviation from the systematic component. The variance of the Laplace distribution needs to be relatively large and as an effect the posterior uncertainty on the regular observations is larger than in the previous methods. Note the similarity to Figure 5.5 for the EP approximation for Gaussian mixture noise. However, here the unimodality is a characteristic of the posterior, whereas in Figure 5.5 it is only due to inability of the approximation to capture the multimodality of the posterior.

as used in Support Vector Regression, which is zero for residuals smaller than ϵ and linear in the absolute value of the residual otherwise. Interpreted as the negative log likelihood of a probabilistic noise model

$$p(y|f(\mathbf{x}), \epsilon) \;\propto\; \exp(-c\,|y - f(\mathbf{x})|_{\epsilon}) \tag{5.53}$$

the probability density of errors is uniform over 2ϵ and then falls off like a Laplace distribution (Chu et al., 2004). In Support Vector Regression the sum of the ϵ-insensitive loss and a regularization term is minimised (Vapnik, 1999, ch. 6). For details on the connection between Support Vector Regression and frequentist robust estimators the reader is referred to Schölkopf and Smola (2002, ch. 9).

We now turn to the question how approximate Bayesian inference can be implemented for the Gaussian process regression model with Laplace noise. Since the posterior is a unimodal distribution one could think of implementing a Laplace approximation by making a second order Taylor approximation around the mode of the unnormalised log-posterior. However, due to the discontinuous derivatives of the likelihood (5.48) at zero, the posterior is also likely to have a kink at its maximum, see Figure 5.11 for an illustration. Therefore, a Laplace approximation of the posterior is inappropriate since the maximum of the posterior and the kink can coincide in which case the Hessian is undefined. Markov chain Monte Carlo sampling can be implemented analogously to Student-t noise described in Section 5.4.2, using the scale-mixture representation (5.49).

Alternatively an Expectation Propagation approximation as described in Section 4.2 can be implemented and the necessary moments (4.22):

$$m_k = \int f_i^k \,\text{Laplace}(y|f_i, s)\,\mathcal{N}(f_i|\mu_{\setminus i}, \sigma^2_{\setminus i})\, df_i \tag{5.54}$$

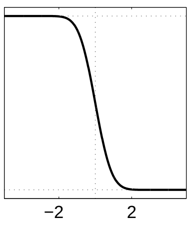


Figure 5.13.: The erfc function.

can be computed analytically for $k = 0, 1, 2$. However, in practice is has shown to be non-trivial to implement the computation of moments in a numerically stable way. Computing the moments naïvely requires to evaluate the complementary error function erfc for large argument values, where the function is very close to zero and numerical approximations fail to give the necessary accuracy, see Figure 5.13.

Although we cannot avoid this problem completely, we can rewrite the expressions in order to obtain numerically more stable expressions later. Therefore, the variables are linearly transformed first

$$\tilde{\mu}_{\setminus i} = (\mu_{\setminus i} - y)/s \tag{5.55a}$$
$$\tilde{\sigma}^2_{\setminus i} = \sigma^2_{\setminus i}/s^2 \tag{5.55b}$$

such that the mean of the Laplace distribution is at zero. In a next step the moments of the transformed variable are computed. Therefore the following two quantities will be needed

$$a = \text{erfc}\left(\frac{\tilde{\sigma}^2_{\setminus i} - \tilde{\mu}_{\setminus i}}{\sqrt{2\tilde{\sigma}^2_{\setminus i}}}\right) \qquad \text{and} \qquad b = \exp\left(2\tilde{\mu}_{\setminus i}\right)\text{erfc}\left(\frac{\tilde{\sigma}^2_{\setminus i} + \tilde{\mu}_{\setminus i}}{\sqrt{2\tilde{\sigma}^2_{\setminus i}}}\right) \tag{5.56}$$

where $\text{erfc}(z) = 1 - \text{erf}(z)$ is the complementary error function. The error function erf is defined as the integral

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z \exp(-t^2)\, dt \tag{5.57}$$

which is approximated numerically (Abramowitz and Stegun, 1965, ch. 7). For large positive values of z the erfc function is very close to zero and standard numerical routines give insufficient accuracy to compute the moments. In our numerical implementation we compute $\ln a$ and $\ln b$ using the approximation

$$\ln \text{erfc}(z) \approx \ln\left(\frac{2}{\sqrt{\pi}}\right) - z^2 - \ln\left(z + \sqrt{z^2 + \frac{4}{\pi}}\right), \tag{5.58}$$

for large values of z (for all $z > 0$ the right hand side is an upper bound on $\ln \text{erfc}(z)$). The moments of the transformed problem can be computed according to

$$\tilde{m}_0 = \frac{a+b}{4}\exp\left(\tfrac{1}{2}\tilde{\sigma}^2_{\setminus i} - \tilde{\mu}_{\setminus i}\right) \tag{5.59a}$$
$$\tilde{m}_1 = \frac{1}{a+b}\left(a\left(\tilde{\mu}_{\setminus i} - \tilde{\sigma}^2_{\setminus i}\right) + b\left(\tilde{\mu}_{\setminus i} + \tilde{\sigma}^2_{\setminus i}\right)\right) \tag{5.59b}$$

and

$$\begin{aligned}\tilde{m}_2 &= \frac{a}{a+b}\left(\tilde{\sigma}^2_{\setminus i} + \left(\tilde{\mu}_{\setminus i} - \tilde{\sigma}^2_{\setminus i}\right)^2\right) + \frac{b}{a+b}\left(\tilde{\sigma}^2_{\setminus i} + \left(\tilde{\mu}_{\setminus i} + \tilde{\sigma}^2_{\setminus i}\right)^2\right) \\ &\quad -\frac{2}{a+b}\exp\left(-\frac{1}{2}\left(\tilde{\sigma}^2_{\setminus i} + \frac{\tilde{\mu}^2_{\setminus i}}{\tilde{\sigma}^2_{\setminus i}} - \ln\left(\frac{2}{\pi}\right)\right) + \tilde{\mu}_{\setminus i} + \frac{3}{2}\ln\tilde{\sigma}^2_{\setminus i}\right)\end{aligned} \tag{5.59c}$$

which have to be implemented in a numerically safe way by handling logarithmic values wherever possible. Finally, moments of the original problem (5.54) are

$$m_0 = \tilde{m}_0/s \tag{5.60a}$$

$$m_1 = s\tilde{m}_1 + y \tag{5.60b}$$

$$m_2 = \tilde{m}_2 s^2 + 2y\tilde{m}_1 s + y^2 \tag{5.60c}$$

which corrects for the transformation (5.55). Figure 5.11(c) gives an illustration of the posterior and its EP approximation in one dimension.

In various experiments we observed that the EP approximation for the Gaussian process regression model with Laplace noise converges more reliably than the EP approximation for the mixture noise model. To avoid numerical problems it is important to scale the targets $\mathbf{y}$ such that the moments (5.59) can be computed in a numerically stable range of values. In our experiments it was sufficient to scale the targets to zero mean and unit variance.

A variational approximation as described for Student-t noise cannot be implemented analogously since if the noise variance σ^2 has an exponential prior distribution, the expectation of the precision $\mathbb{E}[\sigma^{-2}]$ is not finite.

5.6. Experiments

This section presents experiments comparing the performance of several regression models and approximation techniques. The purpose of the experiments will be twofold. One aspect of interest will be the question whether the proposed robust Gaussian process models lead to improved predictive performance. A second aspect will be the assessment of practical problems of the approximation schemes, e.g. convergence and runtime behaviour.

For all Gaussian process models an anisotropic squared exponential covariance function (3.37) is used with individual length scale parameters ℓ_i for each input dimension $i = 1, \ldots, n$. Including the signal variance σ_s^2, the Gaussian process prior has $n+1$ hyper-parameters $\boldsymbol{\psi} = [\boldsymbol{\ell}, \sigma_s^2]$. Note that using a covariance function of this type makes the model inherently robust with respect to outliers in the inputs $\mathbf{x}$, since observations far away will become uncorrelated to regular observations.

For comparison, the performance of Support Vector Regression (SVR) will be reported. The implementation provided by Chang and Lin (2001) is used with a radial basis function (RBF) kernel, which is equivalent to an isotropic squared exponential covariance function where $\sigma_s^2 = 1$ and one common length scale ℓ, so that all input dimensions are weighted equally. This is a clear disadvantage compared to the anisotropic (ARD) parameterisation we implemented in the Gaussian process models, because the SVR cannot adapt the influence of the input dimensions separately. The SVR algorithm has three parameters, i.e. the insensitivity parameter ϵ, defined in eq. (5.52), a regularization parameter C, and the length scale ℓ, or width, of the RBF kernel. In the experiments values for all three parameters are found by an inner loop of 5-fold cross-validations on the training data. We manually refine the parameter grids and repeat the cross-validation procedure until we find the grid that covers the parameter values for which we observe good performance on the training data, with the empirically best performing values not at the boundary.

The following abbreviations will be used to refer to the different models in the comparison:

- OLS: As a baseline we use ordinary least squares linear regression, as is described for example by Mardia et al. (1979, ch. 6).
- SVR: Support Vector Regression using the implementation provided by Chang and Lin (2001).
- Gaussian process regression with (single) normal noise as described in Section 3.2.
  - snMLII: Values for $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ are found by ML-II estimation.
  - snMCMC: Bayesian inference over $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ is approximated by MCMC sampling as described in the example given in Section 4.3.6.1. We use a wide log-normal prior for the elements of $\boldsymbol{\ell}$ while we use constant priors on all other parameters.
- Gaussian process regression where the sampling distribution of errors is a mixture of two normal distributions as described in Section 5.3.
  - mnEP: Bayesian inference is approximated using Expectation Propagation as described in Section 5.3.1. Values of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ are found by maximising the EP approximation to the evidence.
  - mnLAP: Laplace's method as described in Section 4.1 is used for prediction and approximate ML-II estimation of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$.
  - mnMCMC: Inference is approximated using the MCMC scheme described in Section 5.3.2.
- Gaussian process regression with Student-t noise as described in Section 5.4.
  - tVar: Approximate inference and ML-II estimation is implemented using the variational EM scheme described in Section 5.4.1.
  - tMCMC: MCMC sampling using the implementation described in Section 5.4.2.

- lapEP: Expectation Propagation approximation in the GP regression model with Laplace noise as described in Section 5.5.

Hence, for the GP regression models the prefix encodes the noise model[1] whereas the postfix specifies how Bayesian inference is approximated.

For all methods involving maximum likelihood II estimation, the implementations minimise the negative (approximate) value of the log evidence using a conjugate gradient optimisation routine.

Basically all techniques for approximate Bayesian inference based on Gaussian approximations to the posterior involve non-convex optimisation problems, such that numerical optimisation routines do not necessarily find the global optimum, getting stuck in local optima. Therefore an approximation scheme usually gives different results depending on the initialisation, i.e. the starting point of the optimisation. Examples for non-convex optimisation problems include maximum likelihood II estimation, the localisation of the mode in Laplace's method for multimodal posteriors, and the EM steps in the variational approximation for regression with Student-t noise. Markov chain Monte Carlo sampling gives a non-deterministic answer by definition. Asymptotically the results become independent of the initial state and the parameterisation of the sampling scheme, but for practical data analysis and finite simulations, the accuracy depends on the mixing of the chains and the number of generated samples. In the following experiments each algorithm was run several times on the same data (fold) using different initialisations. For ML-II based methods, the solution showing highest approximate evidence was chosen for prediction on the test data. For MCMC based methods several preliminary simulations were made in order to find good parameter values of the sampling scheme, e.g. the $\boldsymbol{\epsilon}$ for Hybrid MCMC, but the reported results origin from a single last simulation.

In the sampling based Gaussian process methods we perform approximate inference over the elements of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ and so we have to specify prior distributions over their elements respectively. In order to be as fair as possible we used constant, uniform (e.g. $p(\pi) = \mathrm{Beta}(1, 1)$ in mnMCMC) or very broad priors and we believe their influence on the posterior to be negligible relative to the influence of the observed samples.

For comparing the predictive performance of the various models we report the *root mean square error* (RMSE) and the *mean absolute error* (MAE) of the (mean) prediction. For methods providing full predictive distributions the *negative log predictive probability* (NLP) of the test cases will be reported. For artificial data sets these measures will be given for separate test sets, while for real-world data-sets a 10-fold cross-validation will be used (Stone, 1974). In the experiments presented below it will become apparent that the performance of an algorithm can have large variance over the folds in a cross-validation scheme. It can be expected that the variance over the folds is increased if outliers are present in the data set.

Let $\mathbf{X}_*$ again denote test inputs and $\mathbf{t}_*$ the corresponding test targets, either in the form of noise free $\mathbf{f}_*$ or noisy targets $\mathbf{y}_*$ depending on whether the data is synthetic.

[1]Abbreviations of noise models: sn = single normal, mn = mixture noise, t = Student-t, and lap = Laplace.

The *root mean square error* is defined as

$$\text{RMSE} \;=\; \sqrt{\frac{1}{m^*}\sum_{i=1}^{m^*}(t_i^* - \mathbb{E}[f_i^*])^2} \tag{5.61}$$

where m^* denotes the number of test cases. The RMSE can be highly dominated by a few large residuals, so we also report the *mean absolute error*

$$\text{MAE} \;=\; \frac{1}{m^*}\sum_{i=1}^{m^*}|t_i^* - \mathbb{E}[f_i^*]| \tag{5.62}$$

in which the influence of a single error scales linearly. Note that from a decision theoretic point of view the expected absolute loss would be minimised by the median of the predictive distribution. If the predictive distribution of f_i^* is Gaussian, as for the analytic approximations, the mean and the mode coincide and therefore no distinction has to be made. For MCMC methods the predictive distribution is a mixture of Gaussians, in which mode and median can be very different. However, we do not exploit this distinction and report the absolute error of the mean prediction.

Gaussian process models provide predictive distributions for the latent function values $p(\mathbf{f}_*|\mathcal{D}, \mathbf{X}_*, \mathcal{M})$ and including the inferred noise we can compute $p(\mathbf{y}_*|\mathcal{D}, \mathbf{X}_*, \mathcal{M})$, where $\mathcal{M}$ denotes the model. By *negative log predictive probability* we refer to the average negative log value of the predictive distribution

$$\text{NLP} \;=\; -\frac{1}{m_*}\sum_{i=1}^{m_*}\ln p(t_i^*|\mathcal{D}, \mathbf{x}_*, \mathcal{M}) \tag{5.63}$$

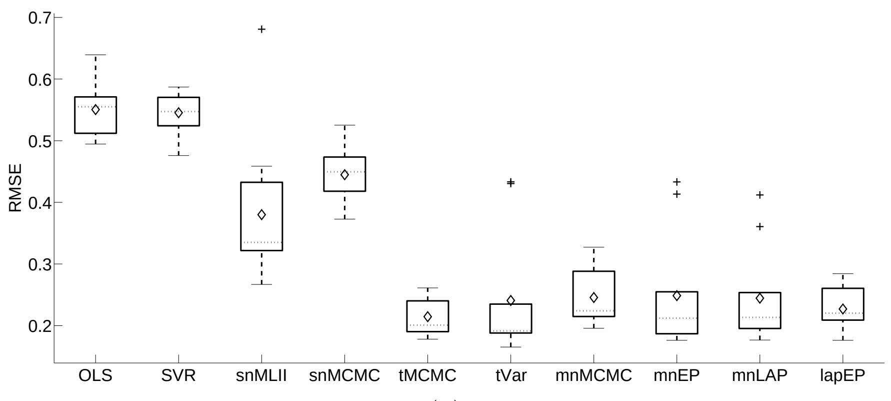
evaluated at the test samples. The NLP is a measure of accuracy of the predictive distribution. Note that NLP are highly non-robust in the sense that a single large error can dominate the estimate. For artificially generated data the test targets $\mathbf{t}_*$ are noise-free function values so we use the predictive distribution $p(\mathbf{f}_*|\mathcal{D}, \mathbf{X}_*, \mathcal{M})$, while for real world data sets only noisy test targets $\mathbf{y}_*$ are given and the respective noise model has to be used to compute the NLP.

5.6.1. Friedman Data

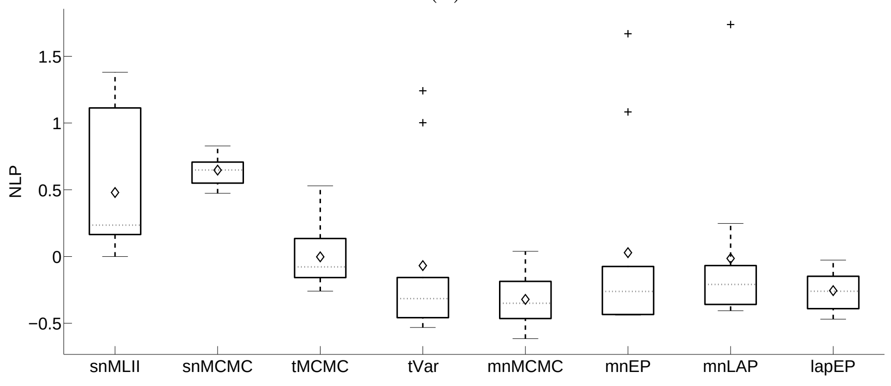
For the first set of experiments an artificial regression problem has been derived from an example described by Friedman (1991). Given 10-dimensional input vectors $\mathbf{x}$ the function value $f(\mathbf{x})$ depends on the first five input dimensions only

$$f(\mathbf{x}) \;=\; 10\sin(\pi\, x_1\, x_2) + 20\,(x_3 - 0.5)^2 + 10x_4 + 5x_5\;, \tag{5.64}$$

while the purpose of the remaining input dimensions $x_6, \ldots, x_{10}$ is only to complicate the task by adding a *feature selection* problem, i.e. identification of the relevant inputs. We generate 10 training sets of $m = 100$ examples respectively. The inputs $\mathbf{x}$ are randomly



(a)



(b)

Figure 5.14.: Performance of various methods on the Friedman data sets. The methods corresponding the abbreviations are described in the beginning of Section 5.6 on page 81. Panel (a) shows box-plots of the root mean square errors on the 10 different data sets. The dotted line marks the median, while the diamond marks the mean of the performance measures. Panel (b) reports the corresponding NLP measures of the methods giving predictive distributions of the noise free targets f_*.

sampled from the uniform distribution on the unit hyper-cube $[0, 1]^{10}$. We then compute the corresponding function values (5.64) and add normal noise with zero mean and unit variance. So far this procedure is identical to the description by Friedman (1991). To generate outliers, 10 randomly selected outputs are replaced in each training set by samples drawn from a normal distribution with mean $\mu = 15$ and variance $\sigma_o^2 = 9$. Thereby outliers are generated which are unrelated to the function (5.64) but are likely to lie in the same range as the function values. For evaluation we generate a test set of $m_* = 10000$ noise-free samples of (5.64). We report NLP values for predicting these noise-free test targets f_*.

Experimental results are summarised in Figure 5.14. The RMSE measures show that the robust Gaussian process regression models perform significantly better on average

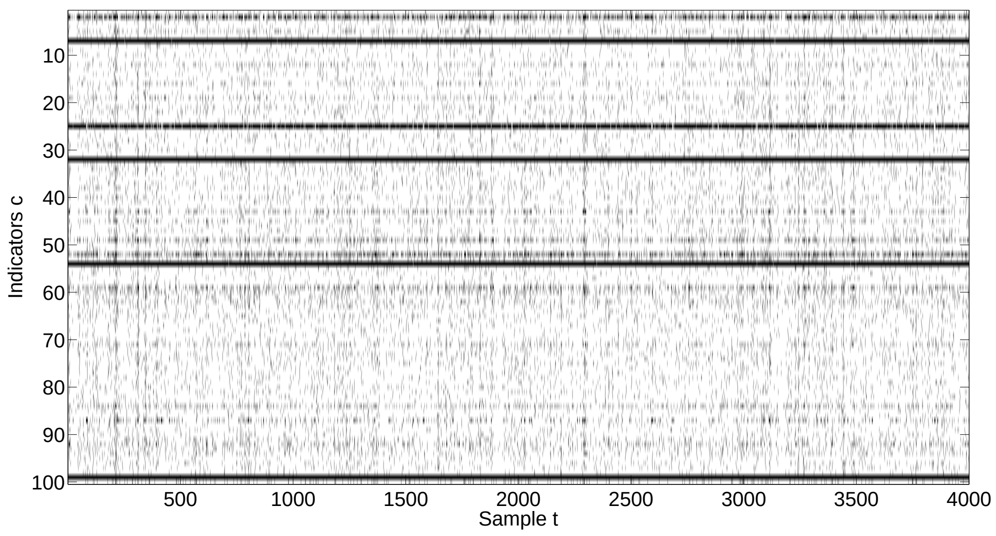


Figure 5.15.: Illustration of indicator variables **c** in mnMCMC. The figure shows 4000 samples of the posterior distribution of **c** as simulated using mnMCMC for one of the Friedman data sets. Each row corresponds to one training observation and a black stripe shows that an observation is attributed to the outlier component. Several observations are considered outliers with high probability, which appear as almost continuous black lines.

than the ones with simple normal noise, which is not surprising given the way the data was generated. The Gaussian process models clearly benefit from their ARD capability, which allows them to ignore the input dimensions which do not prove to be informative about the output. The absence of a similar mechanism may explain the rather poor performance of Support Vector Regression, which performs only slightly better than OLS. The respective mean absolute errors show a very similar pattern and are therefore not discussed separately.

Among the robust GP regression methods tVar, mnEP, and mnLAP show relatively large errors on two folds. Comparing to the respective MCMC results for these noise models (tMCMC and mnMCMC) the poor performance on these two folds must be attributed to the approximation scheme, for example due to local optima problems. In comparison the EP implementation of regression with Laplace noise gives more reliable results. The NLP measures in Figure 5.14(b) give a similar impression. As to be expected, the GP models with simple normal noise also exhibit worse NLP values than the robust variants.

During the experiments none of the implementations showed (unexpected) difficulties. For the mixture noise model the EP approximation did not converge for all values of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. Figure 5.15 shows samples of the posterior on the indicator variables **c** used in the mnMCMC scheme. The method identifies several observations as outliers, which appear as almost continuous black lines.

5.6.2. Boston Housing

The second set of experiments was carried out on the Boston Housing data set. These data have been analysed by Harrison and Rubinfeld (1978) and since then the data has become a popular reference problem in non-linear regression. The task is to predict the median price of houses in different parts of the Boston metropolitan area based on 13 input variables. The target variable appears to be truncated at \$50,000. For a more detailed description, the reader is referred to Harrison and Rubinfeld (1978) or Neal (1996, ch. 4.4.2).

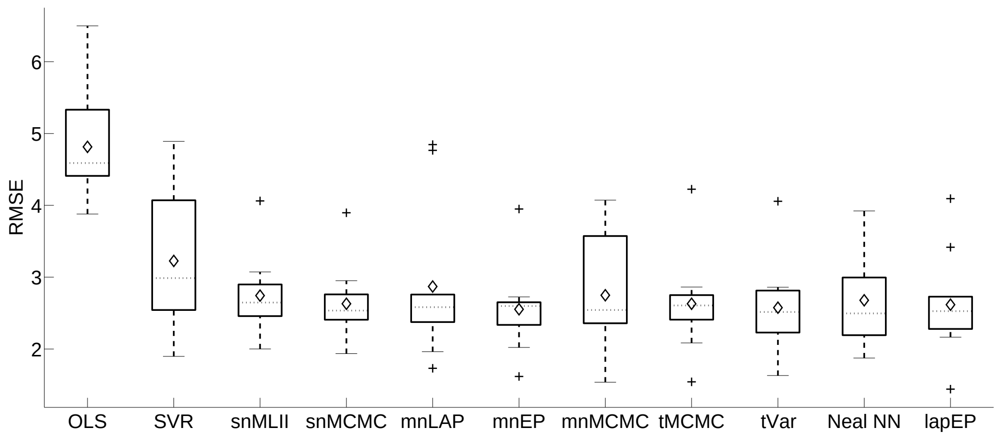
The data set consists of $m = 506$ observations which were normalised to zero mean and unit variance. The data is split into 10 stratified folds, such that the empirical distribution of targets y in each fold is similar to the empirical distribution over the whole sample (Davison and Hinkley, 1997, ch. 3.7). A 10 fold cross-validation procedure is used to estimate the performance of the various methods, which means that each of the folds is left out once as a test set, while the remaining nine folds constitute the training data.

The performance of the various Gaussian process models will be compared to Support Vector Regression and Bayesian neural networks as described by Neal (1996). Apparently, the performance of Bayesian neural networks reported by Neal (1996, ch. 4.4.2) are the best to be found in the literature. Since the Boston Housing data was split differently into folds, we cannot compare to Neal's results directly but used the his software package[2] to replicate the method for our splitting of the data. Following Neal's description a two hidden-layer neural-network was used with six units in each hidden layer. The observational additive noise is modelled by a Student-t distribution with four degrees of freedom. Using the provided software, approximate Bayesian inference is performed over the weights in the neural network using MCMC sampling, i.e. Hybrid Monte Carlo and Gibbs sampling. The exact specification of the network, the priors on the weights, and the sampling scheme are taken from Neal (1996, ch. 4.4.2). After a burn-in period of 150 samples we generated 2000 samples for prediction which took on the order of twelve hours (per fold). Below we use NealNN to refer to this method. Note that the MAE measures computed by Neal's software are mean absolute errors of the *median* prediction, not the mean prediction as in eq. (5.62), which is motivated by a decision theoretic viewpoint as described in Section 2.2.

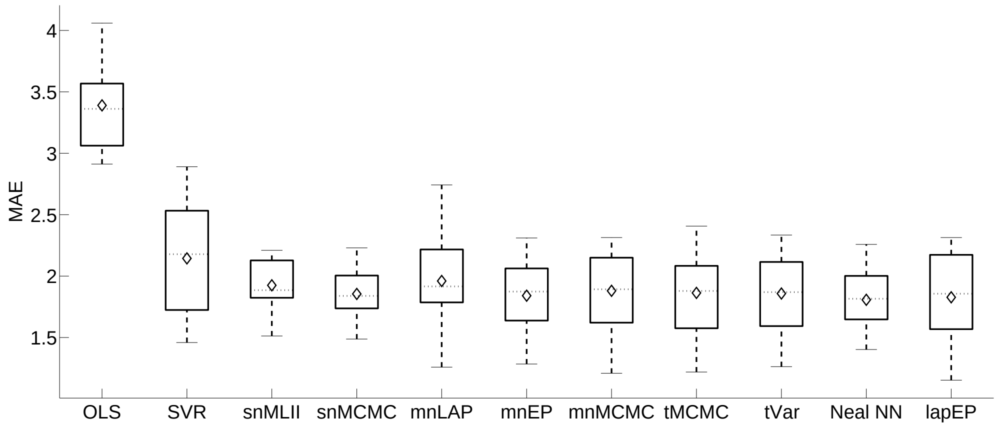
The experimental results are summarised in Figure 5.16 and Table 5.1. All the Gaussian process models and the Bayesian neural networks show relatively similar performance w.r.t. the average RMSE and MAE values.

The lowest average RMSE values are found for mnEP followed by tVar and lapEP. The average RMSE for snMLII is 2.74 compared to 2.55 for mnEP, which is an improvement of 7.5%. Note that the average RMSE for snMCMC is 2.63 which advises caution when attributing the improvement to the noise model only. Support Vector Regression gives worse results on average (RMSE = 3.22) and larger variance over the folds. The Bayesian neural network has a slightly higher average (RMSE = 2.68) which is consid-

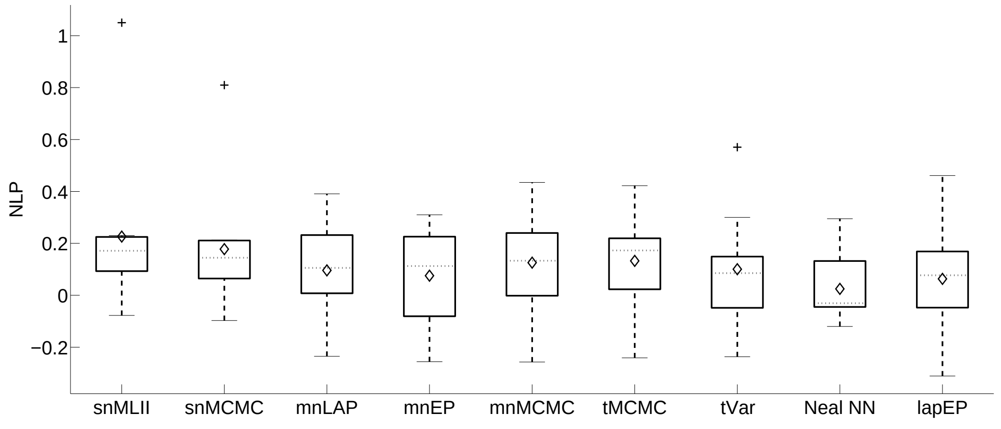
[2] The "Software for Flexible Bayesian Modeling and Markov Chain Sampling" by Radford Neal can be obtained from `http://www.cs.toronto.edu/~radford/fbm.software.html`.



(a)

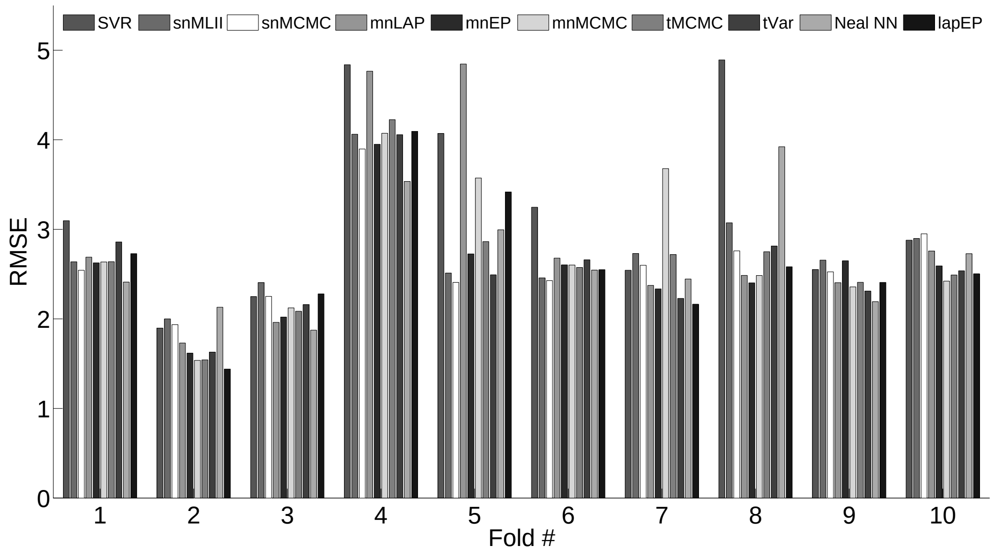


(b)

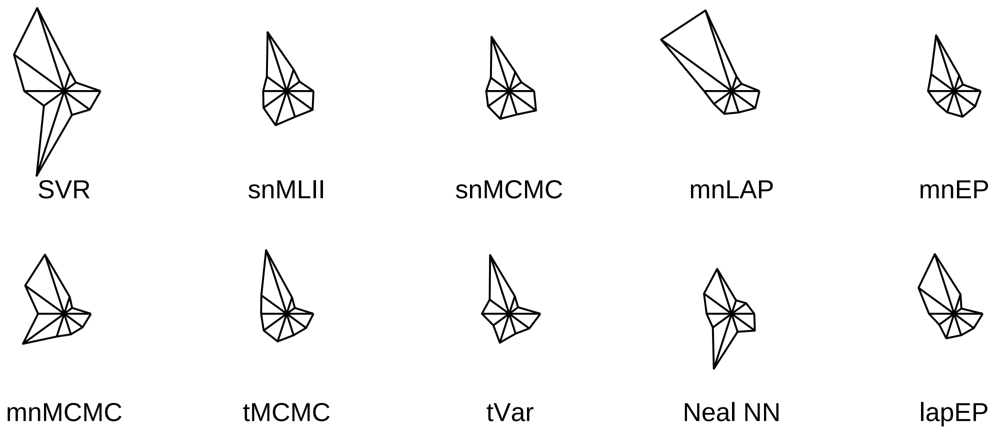


(b)

Figure 5.16.: Performance of various methods on the Boston Housing data. The methods corresponding the abbreviations are described in the beginning of Section 5.6 on page 81. Panel (a) shows box-plots of the root mean square errors on the 10 folds and Panel (b) shows the corresponding mean absolute errors. The dotted line marks the median, while the diamond marks the mean of the performance measures. Panel (b) reports the NLP measures for the methods giving predictive distributions.



(a)



(b)

Figure 5.17.: Fold-wise comparison of the root mean square errors of various methods on the Boston Housing data set. The methods corresponding to the abbreviations are described in the beginning of Section 5.6 on page 81. Panel (a) shows a bar-plot of RMSE for the individual folds. Panel (b) compares the performance pattern of the methods over 10 folds. In this snowflake plot each of the ten rays of a snowflake corresponds to one fold of the Boston Housing data and the magnitude of a ray is proportional to the RMSE on that fold.

Table 5.1.: Average root mean squared error (RMSE), mean absolute error (MAE) and negative log probability (NLP) for various probabilistic regression models and approximation schemes on the Boston Housing data set. The abbreviations are described in Section 5.6 on page 81.

	snMLII	snMC	mnLAP	mnEP	mnMC	tMC	tVar	NealNN	lapEP
RMSE	2.743	2.629	2.869	**2.552**	2.748	2.629	2.574	2.678	2.617
MAE	1.924	1.854	1.960	1.840	1.880	1.863	1.858	**1.807**	1.827
NLP	0.226	0.178	0.096	0.075	0.125	0.132	0.101	**0.025**	0.063

erably worse than the RMSE = 2.49 reported by Neal (1996). This discrepancy can be explained by the different allocation of the data into the folds. A potential mistake in the attempt to set all parameters to exactly the same value as in Neal's simulations can also not be excluded. Furthermore, the results reported by Neal (1996, ch. 4.4.2) were based on 150 posterior samples which were taken of a simulation of length 200. Although we could not reproduce Neal's results, we found the performance measures to have high variance in this experimental setting due to the small number of samples. Using 2000 samples instead stabilised the performance on the folds, but led to slightly worse values. Nevertheless, the Bayesian neural network performed very well in terms of MAE and NLP, also showing less variance over the folds compared to the robust GP models.

Breaking the results down to the individual folds in Figure 5.17 we cannot observe a regular pattern anymore. Fold #4 stands out showing the highest error for almost all methods. Inspecting the Markov chains of several mnMCMC simulations we observed that—especially for folds #5 and #7—the chains had not mixed properly, i.e. the state of the chain did not travel the support of the posterior evenly but rather infrequently switched between discrete areas. This behaviour indicates the presence of local modes of the posterior. We believe that this behaviour did not occur in the previous example because the problems were such that one hypothesis clearly displaced the alternatives. For the folds #5 and #7 the model shows uncertainty whether a few training samples with maximum (truncated) target value should be fitted as regular observations or should be ignored as outliers. Therefore the model is cautious to predict large y-values for test data which leads to a tendency to underestimate the value for test cases which indeed have maximum target value.

Neal's Bayesian neural networks shows a different pattern of errors. While the method performs very well on most folds, larger errors are observed on folds #2 and #7. Since the noise is also assumed to be Student-t distributed, the advantage of this model may be the non-stationary neural network prior.

The simulation time per fold in the cross-validation for mnMCMC and mnLAP was in the order of twelve hours. For mnEP the ML-II parameter optimisations converged slowly and the runtime per fold was similar to mnMCMC. The fraction of outliers π inferred by mnEP and mnMCMC lies in the range of $2\% - 4\%$ for all folds and for σ_o^2 we obtain values that are an order of magnitude larger than σ_r^2. The methods mnLAP and tVar were considerably faster with running times of a few hours per fold.

5.6.3. Remote Sensing Data

The third set of experiments were conducted on artificially generated data from a model of the medium resolution imaging spectrometer (MERIS) on board of the Envisat satellite, and in particular the spectral behaviour of chlorophyll concentration in subsurface waters.[3] Eight channels in the visible range ($412nm - 681nm$) where selected, following the work described by Cipollini et al. (2001). The range of variation of the chlorophyll concentration is from 0.02 to 25 mg/m^3. Note that in the experiments presented below the data will be used as a benchmark problem only. The focus will be to show differences between regression models and not to analyse the data as one should do in a careful Bayesian analysis of remote sensing data. The data has been chosen as an example because analysing the data has shown to be numerically challenging, showing larger differences between the methods.

The total number of samples (pairs of *in situ* concentrations and radiances) available for the experiments was equal to 5000. These samples were divided into 10 folds using the same stratification method as described for the Boston Housing data. Both inputs and outputs were standardised to zero mean and unit variance. Furthermore, a linear model is subtracted from the outputs y. Since the radiances are highly correlated the inputs were projected onto the first five principle components. This rather arbitrary transformation of the data was necessary to make the regression models with normal noise work on the data. Screening the original data no obvious outliers can be detected (see Figure 5.18), but using a Gaussian process regression model with normal noise gives very poor predictions and extreme ML-II estimates of the parameters, especially of the signal variance. Note that this indicates severe model mismatch, which could be with respect to both the noise model and the GP prior.

Results are shown in Figure 5.19. The GP regression models with normal noise show large errors and are clearly inferior to the robust models. For the Student-t model we observe that tMCMC gives more accurate predictions than the variational approximation tVar. The regression model with Laplace noise shows lowest variance over the folds and low RMSE values. The NLP measures for tVar show that on several folds the predictive distributions were inaccurate. Although the median NLP is relatively small, there are three folds on which tVar shows poor performance. Inspecting the folds we see that for a few test examples the predictions are over-confident, which highly affects the NLP measure. The reason behind this over-confidence could either be overfitting or a problem related to local minima.

5.7. Conclusions & Discussion

In applied regression analysis the potential existence of outliers in the data can rarely be ruled out with certainty. In this situation the analyst should choose a model which takes

[3] I would like to thank Joaquin Quiñonero-Candela, Gustavo Camps-Valls, and Paolo Cipollini for kindly providing the data.

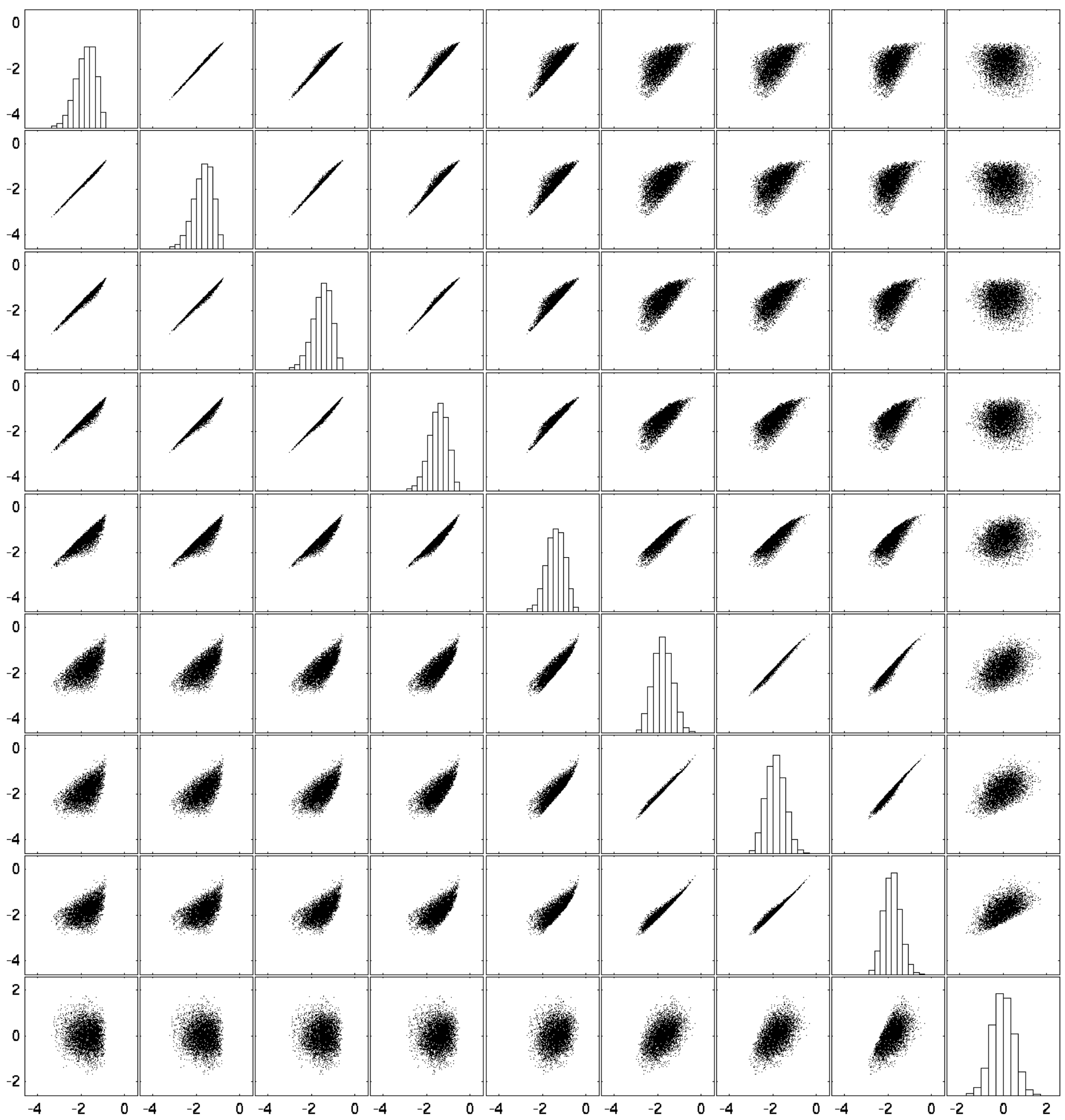
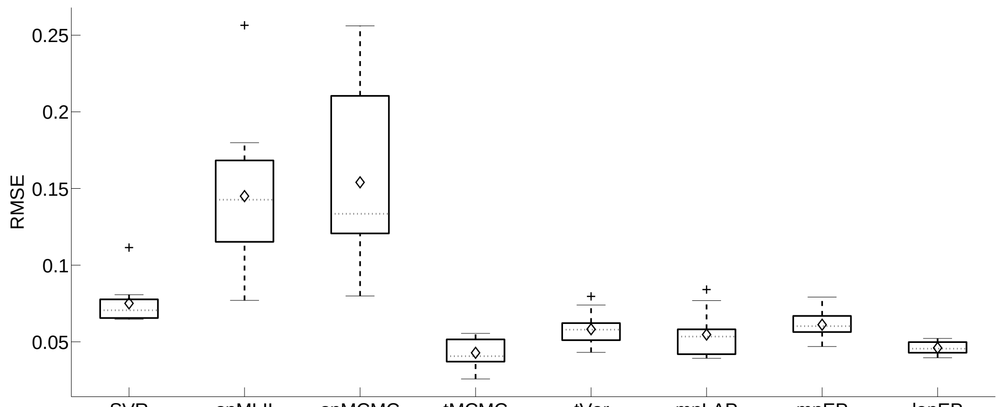
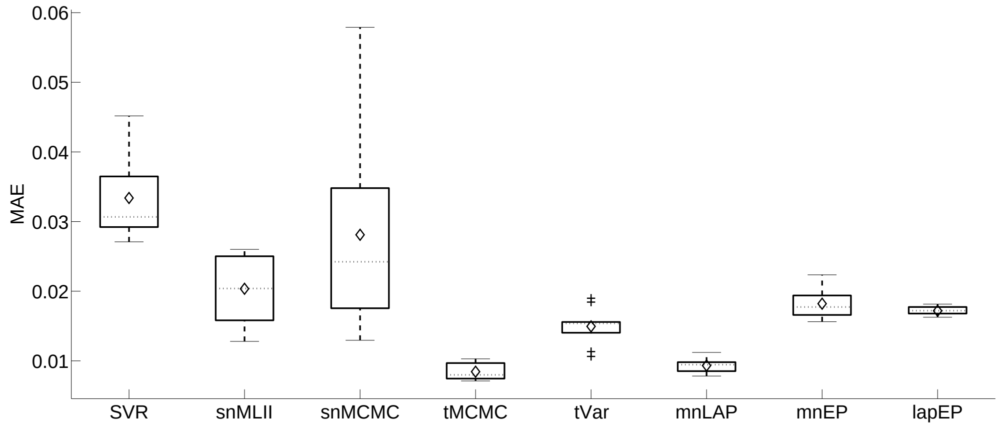


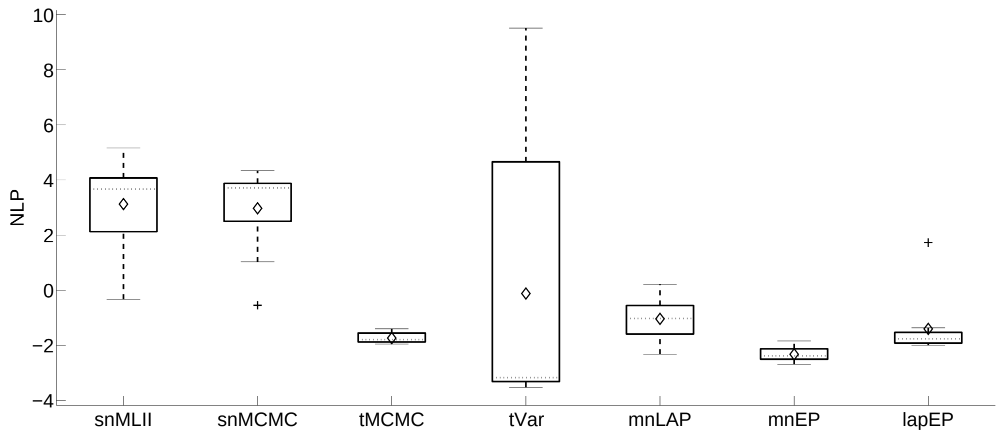
Figure 5.18.: Illustration of the MERIS remote sensing data by pairwise scatter-plots. The first eight columns/rows correspond to the inputs (radiances) and the final column/row are the targets (chlorophyll concentrations). Histograms of the respective measurements are plotted on the diagonal.



(a)



(b)



(b)

Figure 5.19.: Performance of various regression methods on the MERIS remote sensing data. The abbreviations are described in Section 5.6 on page 81. Panel (a) shows box-plots of the root mean square errors on the 10 folds and Panel (b) shows the corresponding mean absolute errors and (c) the NLP measures for methods that provide a predictive distribution. The dotted line marks the median, while the diamond marks the mean of the performance measures.

this belief explicitly into account. In this chapter several variants of Gaussian process models for robust regression were proposed and described how approximate Bayesian inference can be implemented.

Robust regression models are constructed by using a heavy-tailed distribution to model the sampling distribution of errors. In this chapter the use of a mixture noise model, Student-t noise, and Laplace noise has been studied. Markov chain Monte Carlo sampling can be used for approximate inference in all of the proposed models. For GP regression with Laplace noise and the Gaussian mixture noise model it has been described how the EP approximation can be implemented. For regression with Student-t noise a variational approximation has been described.

In the experiments the performance of the robust regression models and several other regression techniques have been compared on three data sets. Experiments on artificially generated data show that the robust Gaussian process regression models outperform the other models in this comparison when outliers in y are present in the training data. In terms of RMSE the performance of GP models on the Boston Housing data set could not be improved by more than 7.5%. Since the target variable is the *median* of house prices in a given area the presence of extreme outliers also seems unlikely. There are arguments suggesting that the noise is not Gaussian, but whether outliers in y are present is unclear. The remote sensing data set has shown to be a difficult problem for Gaussian process regression models. While above the data was used just as a benchmark problem, it would be interesting to study what exactly makes this data difficult, e.g. whether other types of covariance functions would be better suited to explain the data.

All noise models proposed in this chapter have their respective strengths and weaknesses. The normal noise model is convenient, because the posterior and the evidence can be computed in closed form. But the assumption of normal noise makes the model very sensitive to outliers in the data. Using a finite mixture of normal distributions gives a more robust model but inference becomes analytically intractable and difficult to approximate. The finite mixture approach has the conceptual advantage that the outliers do not affect the uncertainty on regular observations, since they are modelled separately. Assuming a Student-t distribution is appealing since its density has very heavy tails. As for the finite mixture noise model, the resulting posterior can be multimodal, which makes approximate Bayesian inference technically challenging. This is avoided by using the Laplace noise model, for which the resulting posterior is unimodal, while the noise distribution is more kurtotic than the normal distribution. See Figure 5.20 for a graphical comparison.

In the Student-t and the mixture noise GP models the posterior distribution can be multimodal. Intuitively, the posterior is multimodal if the model can explain the data in various distinct but equally plausible ways. In this case the approximation by a single Gaussian, as in mnEP or tVar, can be inappropriate and also MCMC sampling is difficult. For simpler outlier-structures this problem might be negligible, since the posterior can be expected to have a strongly dominant mode.

However, most methods used in the comparisons involve either non-convex optimisation problems, sampling from multimodal posteriors or iterative algorithms that do not converge reliably. This introduces a certain difficulty when comparing them, because

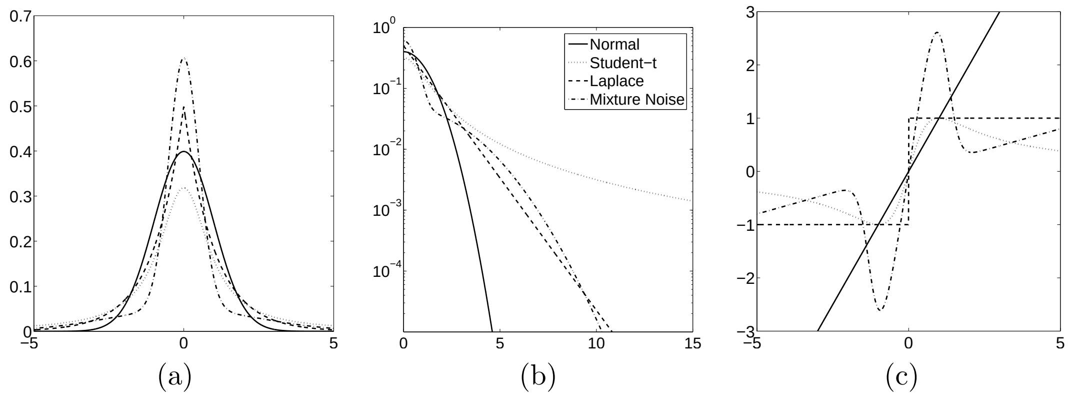


Figure 5.20.: Comparison of noise models. Panel (a) depicts the probability density functions of the normal, Student-t, Laplace and a mixture of two normal distributions. To emphasise the different behaviours in the tails, Panel (b) shows the log of the probability density functions. Clearly normal noise and Laplace noise show to be log-concave. Panel (c) shows Huber's ψ functions ($\psi(x) = -\partial \ln p(x)/\partial x$) for the four noise models.

the observed performances of the methods will depend on the respective initialisation of parameters or the length of MCMC simulations. For the analytic approximations (snMLII, mnLAP, mnEP, tVar, lapEP) several restarts were made and results were reported for the solution showing highest (approximate) evidence. The comparisons are fair (and realistic) in the sense that each method was given only a limited amount of runtime, which was on the order of several hours per fold for the Boston Housing and the remote sensing examples.

MCMC sampling in the mixture noise model (mnMCMC) was the computationally most demanding method in the comparison. A conceptual advantage of MCMC methods is that inference is performed over the function and all parameters jointly. In sampling methods problems related to multimodal posterior distributions can be alleviated by running several shorter chains from different initial states in favour of a single long chain, which in finite simulation can get stuck in a local mode. However, setting the parameters in MCMC sampling schemes and inspecting the chains requires some experience.

Approximate inference by Expectation Propagation iterates between approximate inference over the latent function and ML-II estimation of the remaining parameters. Our implementation suffers from convergence problems in two ways. The first is that for given parameters EP might not converge. We observed that its convergence behaviour highly depends on the values of the parameters, especially for the mixture noise model (mnEP). The second problem is inherent to the ML-II parameter estimation, where our gradient ascent method can get caught in local maxima. Doing multiple runs of the algorithm on the same data sets the respective values of the approximate evidence, however, provide a reliable indication as to which solution to choose. Note that for large data sets, e.g. $N \gg 1000$, one could explore sparse EP approximations to the posterior process following the lines of Csató and Opper (2002) or Lawrence et al. (2003).

The variational approximation for the regression model with Student-t noise has shown to work relatively well in the experiments, although convergence of $\mathcal{V}$ can be slow and the algorithm can converge to sub-optimal local maxima. An advantage of this approximation is that it is relatively easy to implement and it does not require to set additional parameters. However, it is difficult to characterise properties of the approximation relative to the true posterior.

The Laplace noise model is advantageous from a computational point of view, since the posterior is unimodal and thereby easier to approximate. In practice the Laplace noise model may give a reasonable compromise between outlier robustness and computational reliability.

In summary, this chapter introduced and compared several robust models for Gaussian process regressions. These models constitute a practical way of implementing Gaussian process regression in situations in which the potential existence of outliers in the data cannot be ruled out. But the increase in robustness comes with the price of a significant increase in computational complexity and implementing approximate Bayesian inference becomes technically challenging.

5.8. Bibliographical Remarks

A large amount of literature on robust Bayesian methods has been published since the middle of the last century, see for example the overview by Berger (1991) and the references therein. An early and influential work is Box and Tiao (1962), who proposed the use of more kurtotic sampling distributions than the normal distribution for the construction of robust models. O'Hagan (1979) discusses the effects of outliers in Bayesian models based on the "outlier-prone" and "outlier-resistant" distributions described by Green (1976). Box (1979) considers robustness from a more conceptual point of view based on the insight that a mismatch between the model and the data-generating process is unavoidable in practical data analysis.

Robust frequentist techniques for location parameter estimation have been introduced by Huber (1964, 1981), developing many useful concepts like the breakdown point and influence functions. Rousseeuw and Leroy (1987) focus on robust estimation in the linear regression model including L_1 regression. Support Vector Regression has been introduced by Drucker et al. (1997) and is described in more detail by Vapnik (1999, ch. 6) and Schölkopf and Smola (2002, ch. 9). Experiments using SVR on the Boston Housing data—in a different experimental setting—can be found in Schölkopf and Smola (2002, ch. 9.6).

The scale-mixture of normal distributions has been introduced by Andrews and Mallows (1974). West (1987) has shown that the scale-mixtures of normal distributions contain the power exponential distributions of Box and Tiao (1962).

The idea of using a mixture of two normal distributions for robust regression can already be found in Box and Tiao (1968). The approach of having separate models for regular and outlier observations is also advocated by Jaynes (2003, ch. 21).

The use of Student-t noise in linear regression models has been described by West (1984) and has been further developed for example by Lange et al. (1989), Geweke (1993), and Fernandez and Steel (1999). In the context of neural network models Student-t noise has been used for instance by Briegel and Tresp (2000) and Neal (1996). The variational approximation described in Section 5.4.1 is based on the work by Tipping and Lawrence (2003, 2005) who describe the variational approximation for robust regression with Student-t noise in the Relevance Vector Machine (RVM) framework introduced by Tipping (2001). See also Faul and Tipping (2001) for a related mixture noise model. More general descriptions of variational methods for approximate Bayesian inference can be found in Jordan et al. (1999), Ghahramani and Beal (2000, 2001), Jaakkola and Jordan (2000), and Beal (2003).

6. Assessing Approximations for Binary Gaussian Process Classification

"What should be done" is almost always more important than "what can be done exactly". — Tukey (1954)

In recent years probabilistic classification models based on Gaussian process priors have attracted much attention in the machine learning community. Whereas inference in the Gaussian process regression model with Gaussian noise can be done analytically (see Section 3.2), probabilistic classification using GPs is analytically intractable. Several approaches to approximate Bayesian inference have been suggested, including Laplace's method, Expectation Propagation, variational approximations, mean field methods, and Markov chain Monte Carlo sampling, some of these in conjunction with generalisation bounds, online learning schemes and sparse approximations (Neal, 1998a; Williams and Barber, 1998; Gibbs and MacKay, 2000; Opper and Winther, 2000; Csató and Opper, 2002; Seeger, 2002).

Despite the abundance of recent work on probabilistic GP classifiers, most experimental studies provide only anecdotal evidence, and no clear picture has yet emerged, as to when and why which algorithm should be preferred. Thus, from a practitioners point of view it remains unresolved which approximation should be used to implement probabilistic Gaussian process classification. This chapter is about understanding and comparing two of the most wide-spread approximations for Gaussian process classification: Laplace's method and Expectation Propagation. The results of these analytic approximations are compared to computationally more demanding MCMC results, which become exact in the limit of long running times. Sparse variants of the EP approximation will not be addressed and the focus will be on comparing the two types of approximation.

Two aspects of the approximation schemes will be examined: Firstly the accuracy of approximations to the evidence which is of central importance for model selection and model comparison. In any practical application of GPs in classification, usually multiple parameters of the covariance function (hyper-parameters $\boldsymbol{\psi}$) have to be handled, while in its basic form the likelihood has no additional parameters. As described in Section 2.3, Bayesian model selection provides a consistent framework for setting such parameters. Therefore, it is essential to evaluate the accuracy of the evidence approximations as a function of the hyper-parameters in order to assess the practical usefulness of the approach. The related question of whether the evidence correlates well with the generalisation performance cannot be answered in general but depends on the appropriateness of the model for a given data set, as described in Section 2.3.2. However, the issue will be assessed empirically for two data-sets.

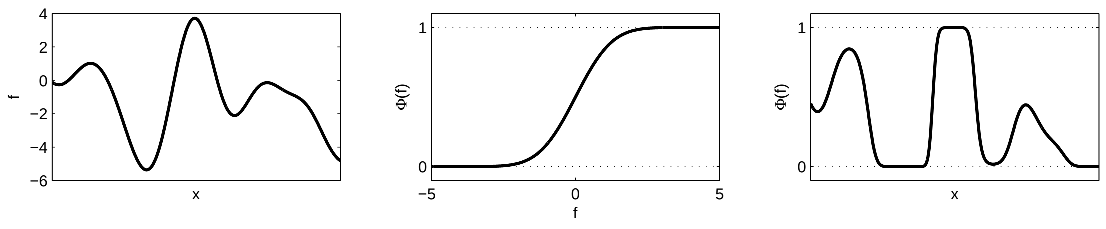


Figure 6.1.: Illustration of the Gaussian process classification model. The left panel depicts an unconstrained latent function $f(x)$ which is mapped to the unit interval by using the probit function (centre panel) which results in $p(y{=}1|x) = \Phi(f(x))$ in the right panel.

Secondly, the quality of the approximate probabilistic predictions will be assessed. In the past, the probabilistic nature of the GP predictions have not received much attention, the focus being mostly on classification error *rates*. This unfortunate state of affairs is caused primarily by typical benchmarking problems being considered outside of a realistic context. The ability of a classifier to produce class probabilities or confidences have obvious relevance in most areas of application, e.g. medical diagnosis and ROC analysis. The predictive distributions of the approximate methods will be evaluated and compared to the ones obtained by extensive MCMC simulations.

6.1. The Gaussian Process Model for Binary Classification

This section describes the Gaussian process model for binary classification (GPC). Let $y \in \{-1, 1\}$ denote the class label corresponding to on an input $\mathbf{x}$. The GPC model is discriminative in the sense that it models $p(y|\mathbf{x})$ which for fixed $\mathbf{x}$ is a Bernoulli distribution. The probability of success $p(y{=}1|\mathbf{x})$ is related to an unconstrained latent function $f(\mathbf{x})$ which is mapped to the unit interval by a sigmoidal transformation. Most common are the *logit*:

$$p(y{=}1|\mathbf{x}) \;=\; \frac{1}{1+\exp(-f(\mathbf{x}))} \tag{6.1}$$

or the *probit* function:

$$p(y{=}1|\mathbf{x}) \;=\; \Phi(f(\mathbf{x})) \tag{6.2}$$

where Φ denotes the cumulative density function of the standard normal distribution such that $\Phi(f(\mathbf{x})) = \frac{1}{2}(1 + \operatorname{erf}(f(\mathbf{x})/\sqrt{2}))$. For reasons of analytic convenience (for the EP algorithm) only the probit model will be considered in the following, see Figure 6.1.

Given the latent function, the class labels are independent Bernoulli variables, so the joint likelihood factorises:

$$p(\mathbf{y}|f) \;=\; \prod_{i=1}^{m} p(y_i|f(\mathbf{x}_i)) \;=\; p(\mathbf{y}|\mathbf{f}) \tag{6.3}$$

and depends on f only through its value at the observed inputs. For the probit model the individual likelihood terms become

$$p(y_i|f_i) \;=\; \Phi(f_i)^{\frac{y_i+1}{2}} + (1-\Phi(f_i))^{\frac{y_i-1}{2}} \;=\; \Phi(y_i f_i) \tag{6.4}$$

due to the symmetry of Φ. Note that the likelihood has no additional parameters. As prior over functions f we again use a zero-mean Gaussian process prior. The zero mean function encodes that *a priori* $p(y=1|\mathbf{x}) = 1/2$. Note that a non-zero mean could be used to implement a bias towards one of the classes. As described in Section 3.3 the choice of mean and covariance function can be used to encode further beliefs about the characteristics of the latent function.

The posterior distribution over the latent function values $\mathbf{f}$ for given hyper-parameters $\boldsymbol{\psi}$ becomes:

$$p(\mathbf{f}|\mathcal{D},\boldsymbol{\psi}) \;=\; \frac{p(\mathbf{y}|\mathbf{f})\,p(\mathbf{f}|\mathbf{X},\boldsymbol{\psi})}{p(\mathcal{D}|\boldsymbol{\psi})} \;=\; \frac{\mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K})}{p(\mathcal{D}|\boldsymbol{\psi})}\prod_{i=1}^{m}\Phi(y_i f_i) \tag{6.5}$$

which is *unimodal* but non-Gaussian due to the log-concavity of the probit function. Properties of the posterior will be further described in Section 6.2.

The main purpose of classification models is to predict the class label y_* for test inputs $\mathbf{x}_*$. The distribution of the corresponding latent function value $f(\mathbf{x}_*)$ can be computed using eq. (3.5) by averaging over the posterior uncertainty:

$$p(f_*|\mathcal{D},\boldsymbol{\psi},\mathbf{x}_*) \;=\; \int p(f_*|\mathbf{f},\mathbf{X},\mathbf{x}_*,\boldsymbol{\psi})\,p(\mathbf{f}|\mathcal{D},\boldsymbol{\psi})\,d\mathbf{f}, \tag{6.6}$$

and by computing the expectation:

$$p(y_*|\mathcal{D},\boldsymbol{\psi},\mathbf{x}_*) \;=\; \int p(y_*|f_*)\,p(f_*|\mathcal{D},\boldsymbol{\psi},\mathbf{x}_*)\,df_* \tag{6.7}$$

the predictive distribution of y_* is obtained, which is again a Bernoulli distribution. The first term of the right hand side of eq. (6.6) is Gaussian and obtained by conditioning the joint Gaussian prior distribution, which is given by eq. (3.7).

Unfortunately we cannot compute the posterior $p(\mathbf{f}|\mathcal{D},\boldsymbol{\psi})$, the predictive distribution $p(y_* = 1|\mathcal{D},\boldsymbol{\psi},\mathbf{x}_*)$, and the evidence $p(\mathcal{D}|\boldsymbol{\psi})$ analytically, so approximations are needed. As described in Chapter 4 approximations for Gaussian process models are either based on a multivariate normal approximation $q(\mathbf{f}|\mathcal{D},\boldsymbol{\psi}) = \mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A})$ to the posterior $p(\mathbf{f}|\mathcal{D},\boldsymbol{\psi})$ or involve Markov chain Monte Carlo sampling.

Introducing the approximate Gaussian posterior $\mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A})$ into eq. (6.6) gives the approximate posterior predictive distribution $p(f_*|\mathcal{D},\boldsymbol{\psi},\mathbf{x}_*) \approx \mathcal{N}(f_*|\mu_*,\sigma_*^2)$ where $\mu_* = m_*(\mathbf{x})$ and $\sigma_*^2 = k_*(\mathbf{x},\mathbf{x})$ as given by eqs. (4.5). Using this approximation, for the probit likelihood the predictive probability (6.7) of $\mathbf{x}_*$ belonging to class 1 can be computed

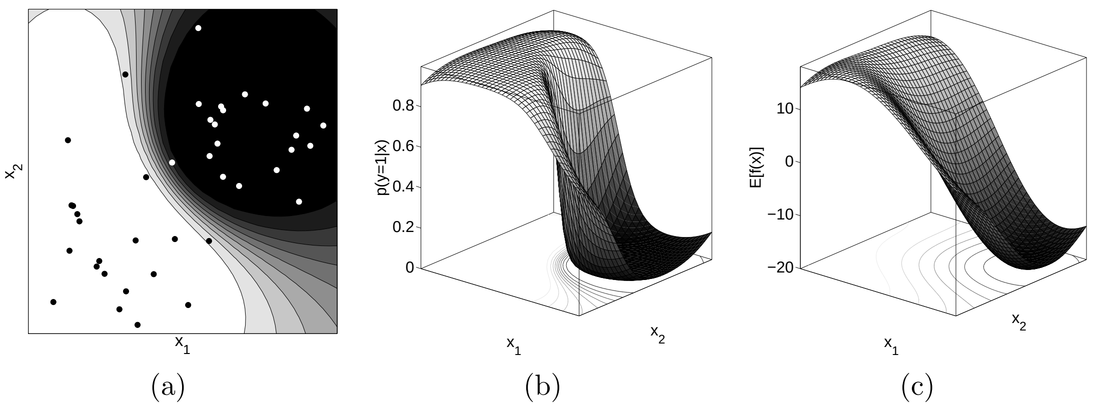


Figure 6.2.: Two dimensional toy example of Gaussian process classification. Panel (a) shows a binary classification problem where black dots correspond to observations of class $y=1$ and white dots are labelled $y=-1$. The contour lines describe the posterior probability of $p(y=1|\mathbf{x})$ which is also plotted in Panel (b). Panel (c) show the posterior mean latent function (found by the EP approximation). The Gaussian process prior uses a squared exponential covariance function (3.36) and the hyper-parameters $\boldsymbol{\psi}$ where selected by maximising the EP approximation to the evidence. Note that away from the observations the mean of the latent function value would be zero, as in the prior, such that the predictive distribution would be $p(y=1|\mathbf{x}) = 1/2$.

analytically:

$$q(y_*=1|\mathcal{D},\boldsymbol{\psi},\mathbf{x}_*) \;=\; \int \Phi(f_*)\,\mathcal{N}(f_*|\mu_*,\sigma_*^2)\,df_* \;=\; \Phi\left(\frac{\mu_*}{\sqrt{1+\sigma_*^2}}\right), \tag{6.8}$$

see Rasmussen and Williams (2006, ch. 3.9) for a derivation. The parameters $\mathbf{m}$ and $\mathbf{A}$ of the multivariate normal approximation to the posterior can be found using Laplace's method or by Expectation Propagation. Both methods also give an approximate value of the marginal likelihood and so ML-II hyper-parameter optimisation can be implemented.

We implemented Laplace's method for the GPC model as described in Section 4.1, details can also be found in Appendix A.2. Since the probit likelihood is log-concave, the posterior is unimodal and the second derivative of the log likelihood is non-positive, such that the $\mathbf{W}$ matrix is always positive semi-definite in eq. (4.10). This ensures that the approximate posterior covariance $\mathbf{A}$ is positive semi-definite in Laplace's approximation, which is in fact always the case for log-concave likelihoods.

In order to implement Expectation Propagation as described in Section 4.2, the zeroth, first, and second moments (4.22) of the product of likelihood and approximate cavity

$$m_k \;=\; \int f_i^k\, p(y_i|f_i)\, q_{\setminus i}(f_i)\, df_i \;=\; \int f_i^k\, \Phi(y_i f_i)\,\mathcal{N}(f_i|\mu_{\setminus i},\sigma_{\setminus i}^2)\, df_i \tag{6.9}$$

have to be computed for $k=0,1,2$. For the probit model these can be found analytically:

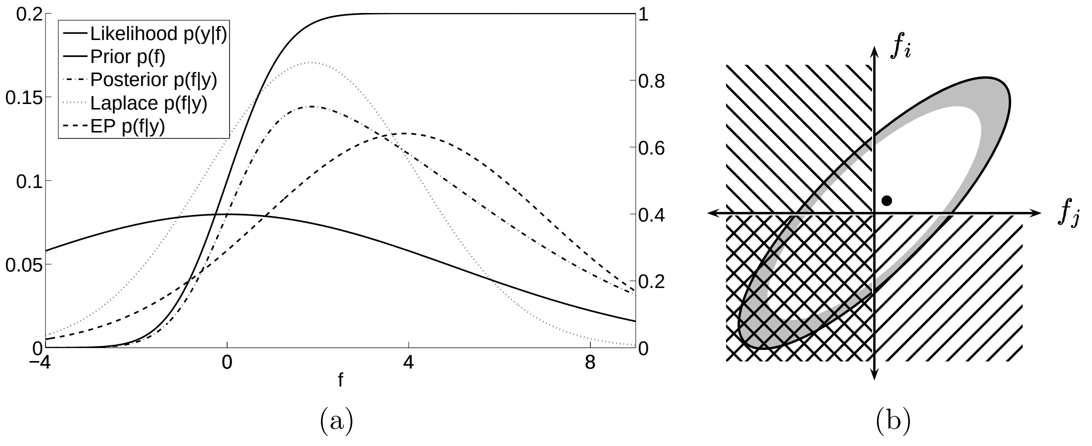


Figure 6.3.: Panel (a) provides a one-dimensional illustration of approximations. The prior $\mathcal{N}(f|0,5^2)$ combined with the probit likelihood ($y=1$) results in a skewed posterior. The likelihood uses the right axis, all other curves use the left axis. In Panel (b) we caricature a high dimensional zero-mean Gaussian prior as an ellipse. The gray shadow indicates that for a high dimensional Gaussian, most of the mass lies in a thin shell. For large latent signals, the likelihood essentially cuts off regions which are incompatible with the training labels (hatched area), leaving the upper right orthant as the posterior. The dot represents the mode of the posterior, which is relatively unaffected by the truncation and remains close to the origin.

$$m_0 = \Phi(z) \quad \text{where} \quad z = \frac{y_i \mu_{\backslash i}}{\sqrt{1+\sigma^2_{\backslash i}}} \tag{6.10a}$$

$$m_1 = \mu_{\backslash i} + \frac{\sigma^2_{\backslash i}\mathcal{N}(z|0,1)}{\Phi(z) y_i \sqrt{1+\sigma^2_{\backslash i}}} \tag{6.10b}$$

$$m_2 = 2\mu_{\backslash i} m_1 - \mu^2_{\backslash i} + \sigma^2_{\backslash i} - \frac{z\sigma^4_{\backslash i}\mathcal{N}(z|0,1)}{\Phi(z)(1+\sigma^2_{\backslash i})}\,. \tag{6.10c}$$

Although the convergence of EP has not been proven formally, for the probit GPC model in various experiments it always converged empirically without exception. Figure 6.2 illustrates the GPC solution for a two dimensional toy example with $m = 40$ observations.

6.2. Structural Properties of the Posterior

The previous sections described the GPC model. This section provides more details on the properties of the posterior which is compared to the structure of Laplace's and the EP approximation.

Figure 6.3(a) provides a one-dimensional illustration. The univariate normal prior combined with the probit likelihood ($y = 1$) results in a skewed posterior. Intuitively, the likelihood cuts off the f values which have the opposite sign of y. The mode of the posterior remains relatively close to the origin, while the mass is placed over positive values in accordance with the observation. Laplace's approximation peaks at the posterior mode, but places far too much mass over negative values of f and too little over large positive values. The EP approximation attempts to match the first two posterior moments, which results in a larger mean and a more accurate placement of probability mass compared to Laplace's approximation.

Structural properties of the posterior in higher dimensions can best be understood by examining its construction. The prior is a correlated m-dimensional Gaussian $\mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$ centred at the origin. Each likelihood term $p(y_i|f_i)$ softly truncates the half-space from the prior that is incompatible with the observed label, see Figure 6.3(b). The resulting posterior is unimodal and skewed, similar to a multivariate Gaussian *truncated* to the orthant containing $\mathbf{y}$. The mode of the posterior remains close to the origin, while the mass is placed in accordance with the observed class labels. Additionally, high dimensional Gaussian distributions exhibit the property that most probability mass is contained in a thin ellipsoidal shell—depending on the covariance structure—away from the mean (MacKay, 2003, ch. 29.2). Intuitively this occurs since in high dimensions the volume grows extremely rapidly with the radius. As an effect the mode becomes less representative (typical) for the prior distribution as the dimension increases. For the GPC posterior this property persists: the mode of the posterior distribution stays relatively close to the origin, still being unrepresentative for the posterior distribution, while the mean moves towards the mass of the posterior making mean and mode differ significantly.

As described, one cannot generally assume the posterior to be close to Gaussian, as in the often studied limit of low-dimensional parametric models with large amounts of data, see for example Schervish (1997, ch. 7.4.2) or O'Hagan (1994, chs. 3.23–3.25). Therefore in GPC one must be aware of making a Gaussian approximation to a non-Gaussian posterior. Laplace's approximation is centred around the mode of the posterior, which lies in the correct orthant but too close to the origin, such that the approximation will overlap with regions having practically zero posterior mass. As an effect the amplitude of the approximate latent posterior GP will be underestimated systematically, leading to overly cautious predictive distributions.

The EP approximation does not rely on a local expansion, but assumes that the marginal distributions of the posterior can be well approximated by Gaussians. As described above, the posterior is similar to a high dimensional multivariate normal distribution truncated to one orthant. Although the marginals of such a distribution are also truncated, they can be surprisingly similar to a Gaussian.

As a low-dimensional illustration Figure 6.4(a-b) shows the marginal distribution of a bivariate normal distribution. Depending on the covariance structure, the mode of the marginal distribution moves away from the origin and the resulting distribution appears similar to a truncated univariate Gaussian.

In order to inspect the marginals of a truncated high-dimensional multivariate normal distribution we made an additional synthetic experiment. We constructed a 767 dimen-

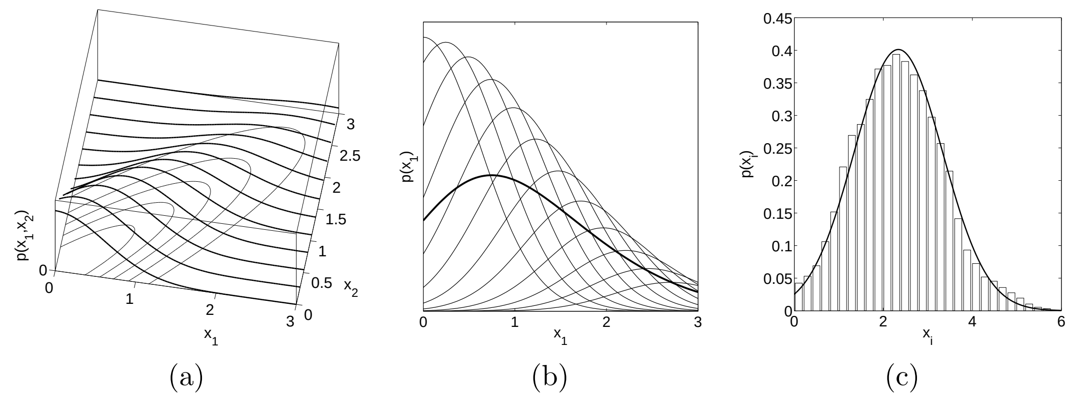


Figure 6.4.: Panel (a) illustrates a bivariate normal distribution truncated to the positive quadrant. The lines describe slices through the probability density function for fixed x_2 values. Panel (b) shows the marginal distribution of $p(x_1)$ (thick line) obtained by (numerical) integration over x_2, which—intuitively speaking—corresponds to an averaging of the slices (thin lines) from Panel (a). Panel (c) shows a histogram of samples of a marginal distribution of a 767-dimensional truncated Gaussian. The line describes a Gaussian with mean and variance estimated from the samples. Note that the particular choice of using a 767-dimensional distribution is motivated by a later experiment shown in Figure 6.10.

sional Gaussian $\mathcal{N}(\mathbf{x}|\mathbf{0},\mathbf{C})$ with a covariance matrix having one eigenvalue of 100 with eigenvector $\mathbf{1}$, and all other eigenvalues are 1. We then truncate this distribution such that all $x_i \geq 0$. Note that the mode of the truncated Gaussian is still at zero, whereas the mean has moved towards the remaining mass. Metropolis-Hastings sampling was used to generate samples from this truncated multivariate distribution. Figure 6.4(c) shows a normalised histogram of samples from a marginal distribution of one x_i. The samples agree very well with a Gaussian approximation. Note that Laplace's method would be completely inappropriate for approximating a truncated multivariate normal distribution.

In order to validate the above arguments we will use Markov chain Monte Carlo methods to generate samples from the GPC posterior and also to estimate the marginal likelihood.

6.3. Experiments

In this section we compare and inspect approximations for GPC using various benchmark data sets. The primary focus is not to optimise the absolute performance of GPC models but to compare the relative accuracy of approximations and to validate the arguments given in Section 6.2.

In all the GPC experiments we use an isotropic squared exponential covariance function of the form:

$$k(\mathbf{x},\mathbf{x}',\boldsymbol{\psi}) \;=\; \sigma_s^2 \exp\big(-\tfrac{1}{2\ell^2}\left\|\mathbf{x}-\mathbf{x}'\right\|^2\big), \tag{6.11}$$

such that $\boldsymbol{\psi} = [\sigma_s^2, \ell]$ where again σ_s^2 denotes the signal variance and ℓ will be referred to as the characteristic length-scale. Note that for many classification tasks it may be reasonable to use an individual length scale parameter for every input dimension (ARD), i.e. an anisotropic covariance function eq. (3.37). Nevertheless, for the sake of presentability we use the above covariance function and we believe the conclusions to be independent of this choice.

Both analytic approximations have a computational complexity which is cubic $\mathcal{O}(m^3)$ as it is common among non-sparse GP models due to inversions of $[m \times m]$ matrices. In our implementations Laplace's method and EP need similar running times, on the order of a few minutes for several hundred data-points.

In order to assess the approximations of the evidence we make use of Annealed Importance Sampling (AIS) as described in Section 4.3.5. Making AIS work efficiently requires some fine-tuning and a single estimate of $p(\mathcal{D}|\boldsymbol{\psi})$ can take several hours for data sets of a few hundred examples, but this could conceivably be improved upon. In all the experiments $\tau(t) = (t/T)^4$ for $t = 0, \dots, 8000$ was used. Using this temperature schedule we found that the sampling spends most of its efforts at temperatures with high variance of the individual ratios (4.45) such that the variance of the resulting estimate (4.46) is relatively small. Note that this was only examined on the data sets used below and only for certain values of $\boldsymbol{\psi}$. In the experiments we combine the estimates of $R = 3$ runs of Thermodynamic Integration to an AIS estimate of the log evidence.

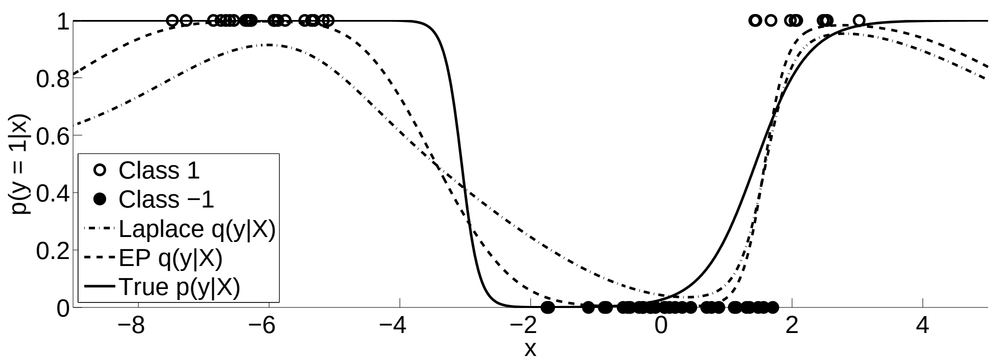
6.3.1. Synthetic Classification Problem

The first experiment is a synthetic classification problem with scalar inputs. The observations for class 1 were generated from two normal distributions with means -6 and 2, each with a standard deviation of 0.8. For class -1 the mean is 0 and the same standard deviation was used.

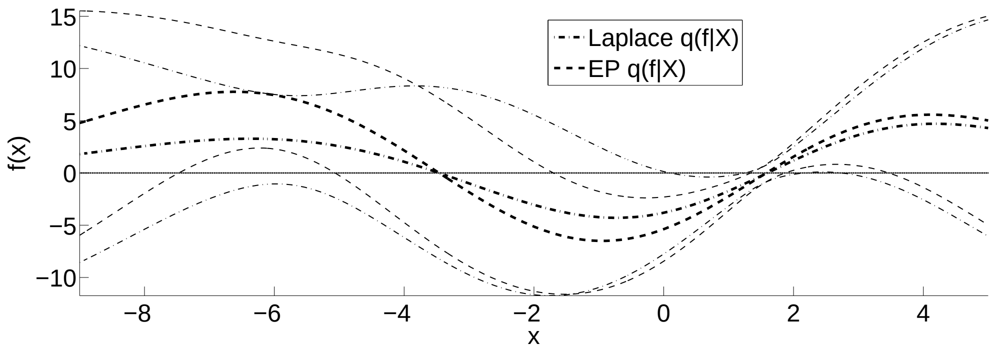
We computed Laplace's and the EP approximation for the ML-II estimated value of $\boldsymbol{\psi}$ that maximised Laplace's approximation to the marginal likelihood (4.12). Note that this particular choice of $\boldsymbol{\psi}$ should be in favour of Laplace's method. Figure 6.5 shows the resulting classifiers and the underlying latent functions. In Figure 6.5(a) the approximations to $p(y|x)$ appear to be similar for positive x but we observe an appreciable discrepancy for negative values. Laplace's approximation gives an unreasonably high predictive uncertainty, which is caused by a significant overlap of the approximate predictive distribution $p(f_*|\mathcal{D}, \boldsymbol{\psi}, \mathbf{x}_*) \approx \mathcal{N}(f_*|\mu_*, \sigma_*^2)$ with zero as shown in Figure 6.5(b). However, note that both approximations agree very accurately on the sign of the predictive mean.

6.3.2. Ionosphere Data

The Ionosphere data origins from a real radar study of free electrons in the ionosphere and was first analysed by Sigillito et al. (1989) using neural networks. The data consists of 351 examples in $n = 34$ dimensions. We standardised the data to zero mean and unit variance. The training set is a random subset of size $m = 200$ leaving the remaining



(a)



(b)

Figure 6.5.: Synthetic classification problem: Panel (a) illustrates the classification task, the generating $p(y=1|x)$ and two approximations thereof obtained by Laplace's method and EP. Panel (b) illustrates the predictive distribution of latent function values showing the mean μ_* and the range of $\pm 2\sigma_*$.

$m_* = 151$ instances out as a test set.

An exhaustive investigation on a regular $[21 \times 21]$ grid of values for the log hyper-parameters is done. For each $\boldsymbol{\psi}$ on the grid the approximated log evidence by Laplace's method (4.12), EP (4.26), and AIS is computed. Additionally the predictive performance on the test set is evaluated. As performance measure we use the average information in bits of the predictions about the test targets in excess of that of random guessing. Let $p^* = p(y_* = 1|\mathbf{x}_*)$ be the model's prediction, then we average:

$$I(p_i^*, y_i) = \frac{y_i+1}{2}\log_2(p_i^*) + \frac{1-y_i}{2}\log_2(1-p_i^*) + 1 \qquad (6.12)$$

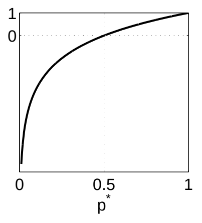


Figure 6.7.: The information (6.12) as a function of p^* for $y=1$.

over all test cases. This measure equals 1 bit if the true label is predicted with absolute certainty, 0 bits for random guessing and takes negative values if the prediction gives higher probability to the wrong class, see also Kononenko and Bratko (1991). Results are shown in Figure 6.6.

For all three approximation techniques we see an agreement between evidence esti-

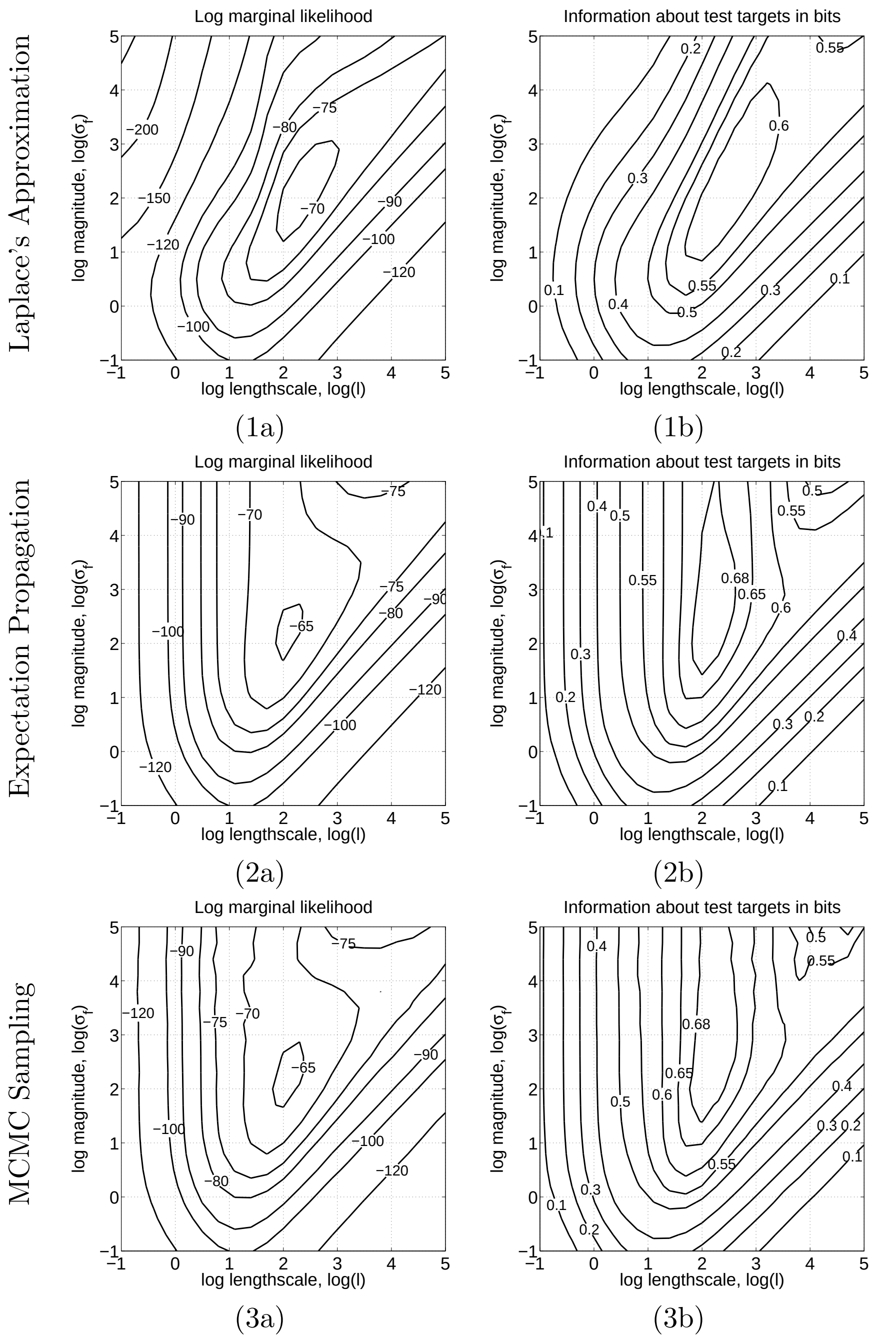


Figure 6.6.: Comparison of marginal likelihood approximations and predictive performances for the ionosphere data set. The first column shows the estimates of log marginal likelihood, while the second column shows the performance on the test set. The first row contains results obtained Laplace's approximation, second row gives the corresponding results for EP and the third row gives MCMC results.

mates and test performance, which justifies the use of ML-II parameter estimation. But the shape of the contours and the values differ between the methods. The contours for Laplace's method appear to be *slanted* compared to EP. However, most remarkable is that the evidence estimates of EP and AIS agree very well.[1] The EP predictions contain as much information about the test cases as the MCMC predictions and significantly more than for Laplace's method.

Note that for small signal variances (roughly $\ln(\sigma_s^2) < 0$) Laplace's method and EP give very similar results. A possible explanation is that for small signal variances the likelihood does not *truncate* the prior but only *down-weights* the tail that disagrees with the observation. As an effect the posterior will be less skewed and both approximations will lead to relatively similar results.

6.3.3. USPS Digits

The USPS digit database contains images of hand-written digits scanned from envelopes and originates from a study described by Hull (1994). Each image is of size $[16 \times 16]$ pixels such that the inputs $\mathbf{x}$ are $n = 256$ dimensional. A binary sub-problem from the USPS digit data is defined by considering the problem of discriminating images showing the digits 3 and 5. Because the training and test partitions in the original data differ significantly, we pooled cases and randomly divided them into new sets, with $m = 767$ cases for training and $m_* = 773$ for testing.

The experiments described in the previous section are repeated for a slightly modified grid of $\boldsymbol{\psi}$ values. Examining the results shown in Figure 6.8 leads to similar conclusions as for the Ionosphere data mentioned above. The EP and MCMC approximations agree very well, given that the evidence comes as a 767 dimensional integral.

We now take a closer look at the approximations $p(\mathbf{f}|\mathcal{D}, \boldsymbol{\psi}) \approx \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{A})$ for a given value of $\boldsymbol{\psi}$. The values $\ln(\sigma) = 3.35$ and $\ln(\ell) = 2.85$ were chosen which are between the ML-II estimates of EP and Laplace's method. Comparing the respective means of the approximations in Figure 6.9(a) we see that the magnitude of the means from the Laplace approximation is much smaller than from EP. The relation appears to be roughly linear. Figure 6.9(b) compares the elements of $\mathbf{W}$ (Laplace's method) and $\boldsymbol{\Sigma}^{-1}$ (EP) which cause the difference in the approximations (4.10) and (4.15) of the posterior covariance matrix $\mathbf{A}$. It shows that the relatively large entries in $\mathbf{W}$ are larger than the corresponding entries in $\boldsymbol{\Sigma}^{-1}$, but in total $\mathbf{W}$ contains more small values than $\boldsymbol{\Sigma}^{-1}$. The exact effect on the posterior covariance is difficult to characterise due to the inversion, but intuitively the smaller the values the more the posterior covariance will be similar to the prior.

Figures 6.9(c–d) compare the predictive uncertainty p^* resulting from the respective approximations to MCMC predictions. For both training and test set we observe that EP and MCMC agree very well, while Laplace's method shows over-conservative (too uncertain) predictions.

[1]Note that the agreement between the two seems to be limited by the accuracy of the AIS runs, as judged by the regularity of the contour lines; the tolerance is less than one unit on a (natural) log scale.

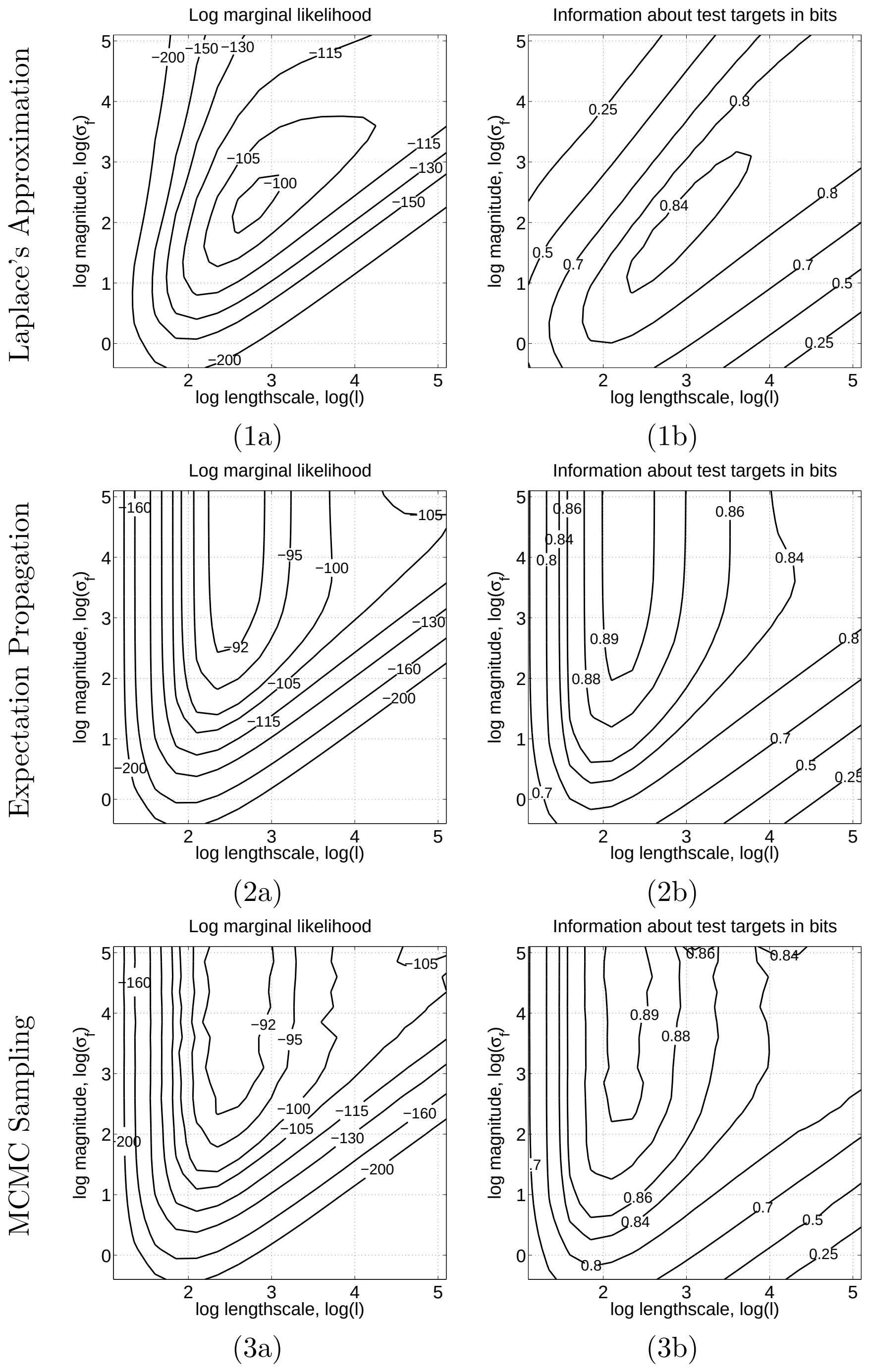


Figure 6.8.: Comparison of evidence (marginal likelihood) approximations and predictive performances of the different methods for classifying 3s vs. 5s from the USPS image database. The first column shows the estimates of log marginal likelihood, while the second column shows the performance on the test set. The first row contains results obtained by Laplace's approximation, second row gives the corresponding results for EP and the third row reports MCMC results.

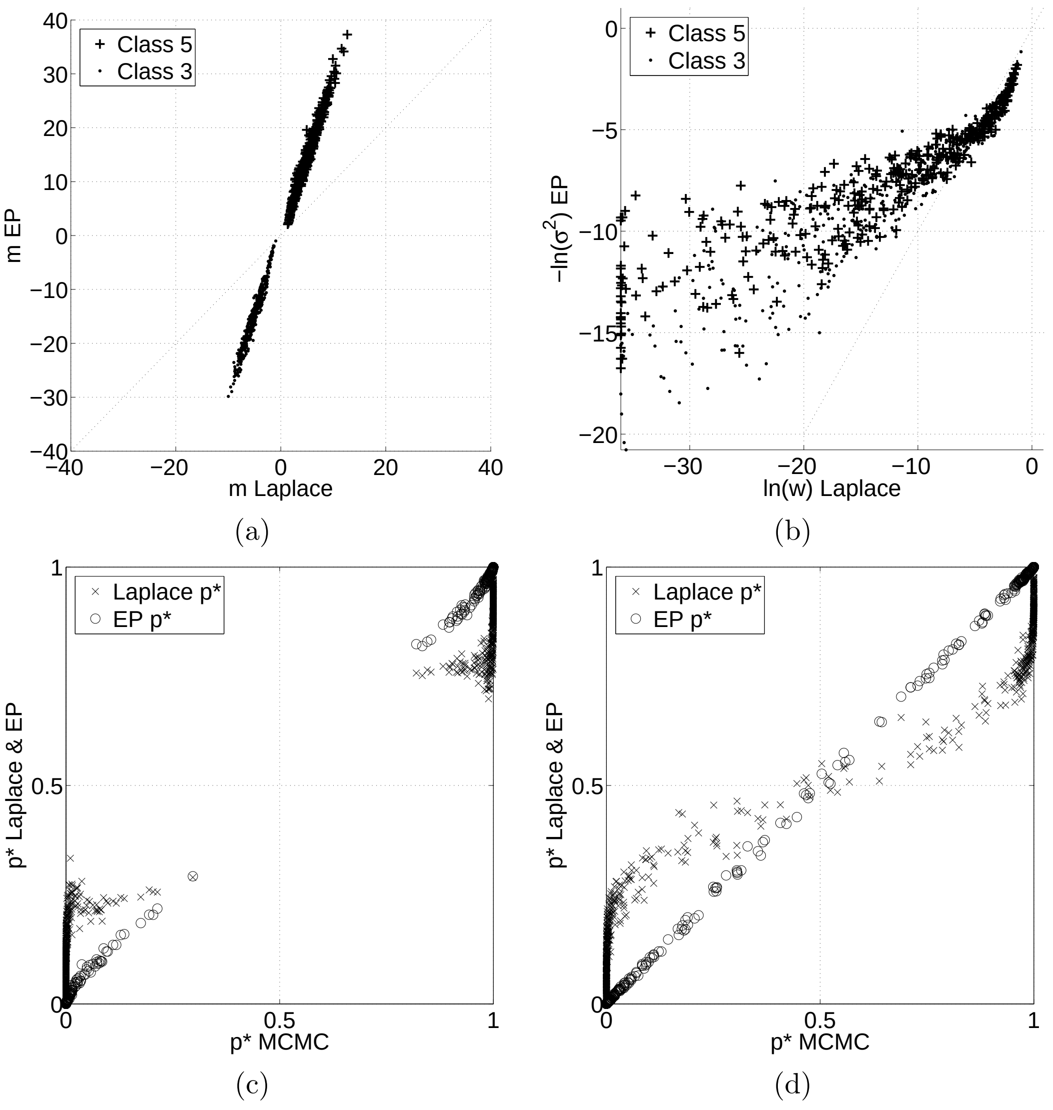


Figure 6.9.: Comparison of approximations $p(\mathbf{f}|\mathcal{D},\boldsymbol{\psi}) \approx \mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A})$ for a given value of $\boldsymbol{\psi}$. Panel (a) shows a comparison of the means m_i. In Panel (b) we compare the elements of the diagonal matrices W_{ii} and Σ_{ii}. Panels (c) and (d) compare predictions p^* obtained by MCMC (abscissa) to predictions obtained from Laplace's method and EP (ordinate). Panel (c) shows predictions on training cases and (d) shows predictions on test cases.

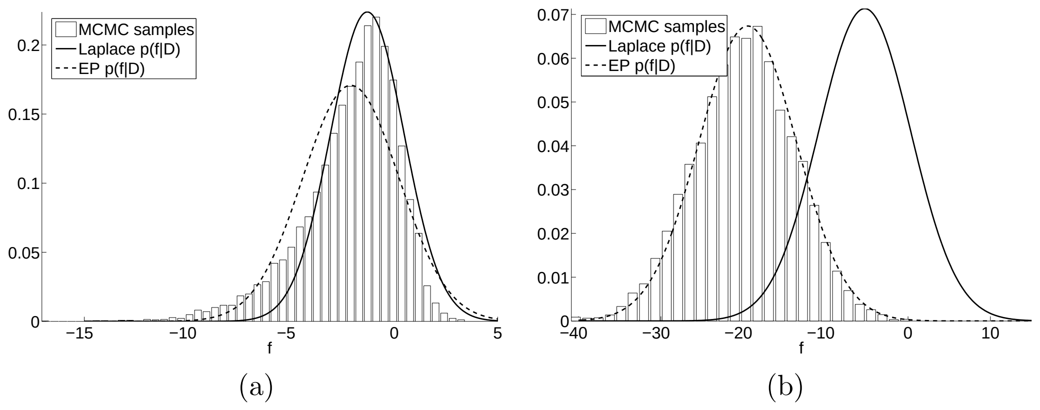


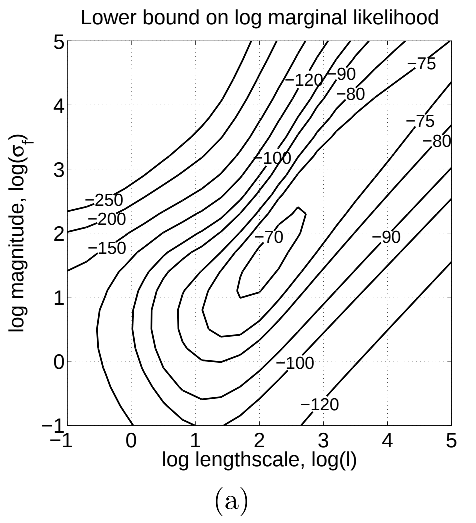
Figure 6.10.: Two marginal distributions $p(f_i|\mathcal{D}, \boldsymbol{\psi})$ from the posterior. For Panel (a) we picked the f_i for which the posterior marginal is maximally skewed (see again Figure 6.3). The true posterior is approximated by a normalised histogram of 9000 samples of f_i obtained by Hybrid Monte Carlo sampling. Panel (b) shows a case where EP and Laplace's approximation differ significantly.

We now inspect the marginal distributions $p(f_i|\mathcal{D}, \boldsymbol{\psi})$ of single latent function values under the posterior approximation. We use Hybrid Monte Carlo to generate 9000 samples from the posterior of $\mathbf{f}$ for the above $\boldsymbol{\psi}$. For Laplace's method and EP the approximated distribution is $\mathcal{N}(f_i|m_i, A_{ii})$ where $\mathbf{m}$ and $\mathbf{A}$ are found by the respective approximation techniques.

In general we observe that the marginal distributions of MCMC samples agree very well with the respective marginal distributions of the EP approximation. For Laplace's approximation we find the mean to be underestimated and the marginal distributions to overlap with zero far more than the EP approximations. Note that this corroborates the conjectures made in Section 6.2 about the structure of the posterior and the properties of the approximations. Figure 6.10(a) displays the marginal distribution and its approximations for which the MCMC samples show maximal skewness. Figure 6.10(b) shows a typical example where the EP approximation agrees very well with the MCMC samples. This particular example was chosen because under the EP approximation $p(y_i = 1|\mathcal{D}, \boldsymbol{\psi}) < 0.1\%$ but Laplace's approximation gives $p(y_i = 1|\mathcal{D}, \boldsymbol{\psi}) \approx 18\%$. In the experiment we saw that the marginal distributions of the posterior typically agree very well with a Gaussian approximation.

6.3.4. A Variational Approximation to the Evidence

In the context of sparse EP approximations Seeger (2003) proposed a lower bound on the evidence. Several authors maximise this lower bound instead of maximising eq. (4.26) for ML-II hyper-parameter estimation with EP approximation (Seeger, 2004; Kim and Ghahramani, 2003; Chu and Ghahramani, 2005). The bound is obtained from the (EP)



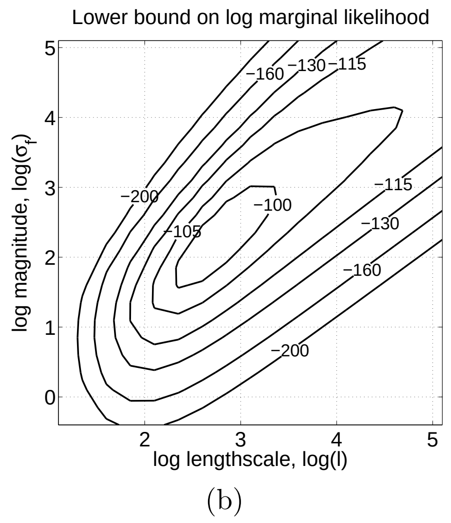


Figure 6.11.: Panel (a) shows the lower bound (6.13c) on the marginal likelihood for the Ionosphere data set (compare to left column of Figure 6.6). Panel (b) shows the value of the lower bound for the USPS 3's vs. 5's (compare to left column of Figure 6.8)
.

approximation of the posterior using Jensen's inequality:

$$\ln p(\mathcal{D}|\boldsymbol{\psi}) = \ln \int p(\mathbf{y}|\mathbf{f})\,\mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K})\,d\mathbf{f} \tag{6.13a}$$

$$\geq \int \mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A})\,\ln \frac{p(\mathbf{y}|\mathbf{f})\,\mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K})}{\mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A})}\,d\mathbf{f} \tag{6.13b}$$

$$= \sum_{i=1}^{m} \int \mathcal{N}(f_i|m_i, A_{ii})\,\ln \Phi(y_i f_j)\,df_i \tag{6.13c}$$
$$-\tfrac{1}{2}\mathbf{m}^\top\mathbf{K}^{-1}\mathbf{m} - \tfrac{1}{2}\,\mathrm{tr}(\mathbf{K}^{-1}\mathbf{A}) + \tfrac{1}{2}\ln\left|\mathbf{K}^{-1}\mathbf{A}\right| + \tfrac{m}{2}\ .$$

Note that the one dimensional integrals in eq. (6.13c) have to be solved using numerical integration methods.

In Figure 6.11 we show the value of the lower bound as a function of the hyperparameters for the Ionosphere and USPS data described in the previous sections. Interestingly, for both data sets the lower bounds appear to be more similar to the approximate evidence obtained by Laplace's method than by EP (compare to Figures 6.6(1a) and 6.8(1a)). However, the maxima of the lower bounds correspond to sub-optimal predictive performances compared to the maxima of the approximate evidence (4.26) (compare to Figures 6.6(2a-b) and 6.8(2a-b)). Therefore for non-sparse EP approximations the use of eq. (4.26) seems advisable, which is also computationally advantageous.

6.3.5. Benchmark Data Sets

In this section we compare the performance of Laplace's method and Expectation Propagation for GPC on several well known benchmark problems for binary classification.

The *Ionosphere*, the *Wisconsin* Breast Cancer, and the *Sonar* data sets are taken from Hettich et al. (1998). The Leptograpsus *Crabs* and the *Pima* Indians Diabetes data sets were described by Ripley (1996). Note that for the Crabs data set we use the sex (not the colour) of the crabs as target variable, by which it becomes a slightly more difficult task. The largest data set in the comparison are the 3's vs. 5's from the USPS hand-written digits database described above.

We standardise the data to zero mean and unit variance. All data sets are randomly split into 10 folds of which one at a time is left out as a test set to measure the predictive performance of a model trained (or selected) on the remaining nine folds.

For GPC we implement model selection by ML-II hyper-parameter estimation. We use a conjugate gradient optimisation routine to find a minimum

$$\boldsymbol{\psi}^\star = \underset{\boldsymbol{\psi}}{\operatorname{argmin}} \; -\ln q(\mathcal{D}|\boldsymbol{\psi}) \tag{6.14}$$

of the negative log evidence approximated by Laplace's method (4.12) and EP (4.26) respectively. For the respective $\boldsymbol{\psi}^\star$ the approximations $\mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{A})$ are computed and predictions are made for the left out test set. From the predictive distributions the average information (6.12) is computed and averaged over the ten folds, i.e. we obtain a cross-validation estimate. Furthermore the average error rate E is reported, which equals the average percentage of erroneous class assignments if prediction is understood as a decision problem with symmetric loss (thresholding the predictive uncertainty at 1/2).

In order to have a better impression of the absolute performance we also report the results of Support Vector Machines (SVMs) (Schölkopf and Smola, 2002). We use the LIBSVM implementation of the soft-margin C-SVM provided by Chang and Lin (2001), with a radial basis function kernel which is equivalent to the covariance function (6.11) up to the signal variance parameter. The values of the length scale parameter ℓ and the regularization parameter C are found by an *inner loop* of 5-fold cross-validation on the nine training folds respectively. We manually refine the parameter grids and repeat the cross-validation procedure until the performance stabilises.

The Support Vector Machine is basically a geometric approach, but we use the technique described by Platt (2000) to estimate predictive probabilities. This is implemented by fitting a sigmoidal mapping from the unthresholded output of the SVM to the unit interval. Let

$$f_{\text{SVM}}(\mathbf{x}) = \sum_{i \in \text{SV}} y_i\, a_i\, k(\mathbf{x}_i, \mathbf{x}) + b \tag{6.15}$$

denote the "unthresholded" output of an SVM, where the sum is over all support vectors. Usually only the sign of $f_{\text{SVM}}(\mathbf{x})$ is used as a point estimate of y_*. Instead, here the idea is to find a mapping of $f(\mathbf{x})$ to the interval $[0,1]$ such that the information (6.12) of the

Table 6.1.: Results for benchmark data sets. The first three columns give the name of the data set, number of observation m and dimension of inputs n. For Laplace's method and EP the table reports the average error rate E, the average information I (6.12) and the average length $\|\mathbf{m}\|$ of the mean vector of the Gaussian approximation. For SVMs the error rate and the average information about the test targets are reported.

			Laplace			EP			SVM	
Data Set	m	n	E	I	$\|\mathbf{m}\|$	E	I	$\|\mathbf{m}\|$	E	I
Ionosphere	351	34	8.84%	0.649	49.96	7.99%	0.719	124.94	**5.69%**	**0.739**
Wisconsin	683	9	**3.21%**	0.870	62.62	**3.21%**	**0.871**	84.95	**3.21%**	0.861
Pima Indians	768	8	22.77%	0.319	29.05	**22.63%**	**0.320**	47.49	23.01%	0.299
Crabs	200	7	**2.0%**	0.682	112.34	**2.0%**	**0.908**	2552.97	**2.0%**	0.047
Sonar	208	60	15.36%	0.443	26.86	13.85%	0.541	15678.55	**11.14%**	**0.571**
USPS	1540	256	2.27%	0.852	163.05	2.21%	0.905	22011.70	**2.01%**	**0.921**

predictions is maximised. Platt (2000) proposes to use a logit model

$$p_*(\mathbf{x}) = \frac{1}{1+\exp(\alpha f_{\text{SVM}}(\mathbf{x})+\beta)} \tag{6.16}$$

and to estimate the parameters α and β by using eq. (6.12) as a likelihood, i.e. the log likelihood of a Bernoulli distribution. Note that for the estimation of α and β a separate data set has to be used, which has been left out from the training data. In our experiments the mapping is estimated on the respective test set in the inner loop of 5-fold cross-validation in which we seek for the parameters of the SVM.

Results are summarised in Table 6.1. Comparing Laplace's method to EP the latter shows to be more accurate both in terms of error rate and information. While the error rates are relatively similar the predictive distribution obtained by EP shows to be more informative about the test targets. As to be expected from previous discussions, the length of the mean vector $\|\mathbf{m}\|$ shows much larger values for the EP approximations. Comparing GPC with EP and SVMs the results are mixed. It could be speculated that the SVM performs better if the input dimension n is large, but in fact the experiments are too few and the results too close to draw any conclusions of this kind.

At first sight it may seem surprising that Laplace's method gives relatively similar error rates compared to EP. Note that for both methods the error rate only depends on the sign of the latent mean function at the test locations, which in turn depend on $\mathbf{m}$ only. Therefore the error rate is less sensitive to the accuracy of the approximation to the posterior, but of course depends on the ML-II estimated hyper-parameters, which differ between the methods. Also in Figure 6.5(b) it can be observed that the latent mean functions differ but their sign matches very accurately.

For the Crabs data set all methods show the same error rate but the information content of the predictive distributions differs dramatically. For some test cases the SVM predicts the wrong class with large certainty. Because the mapping of the unthresholded output of the SVM to the predictive probability is estimated from a left out set, the mapping can be poor if too few errors are observed on this, leading to over-confident predictions.

6.3.6. Excursion on Label Regression

Among all Gaussian process models the regression model with normal noise stands out since the evidence and the posterior process can be computed analytically, as described in Section 3.2. Compared to the binary classification model this makes it a computationally attractive procedure and one could be tempted to *misuse* this model in the classification setting.

Ignoring the discrete nature of the labels $y \in \{-1, 1\}$ one could use a Gaussian process regression model with normal noise and even use ML-II to make point estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. This procedure will be referred to as *label regression* (LR) or *least-squares classification* (Rifkin and Klautau, 2004). For predicting the class label of a test instance y_* a heuristic approach is to use the sign of the posterior mean function

$$y_*(\mathbf{x}) = \operatorname{sign}(m_*(\mathbf{x})) \tag{6.17}$$

where $m_*(\mathbf{x})$ is given by eq. (3.12a). Note that one could also use a similar method as proposed by Platt (2000) for the Support Vector Machine to estimate a kind of predictive uncertainty of y_*.

From a Bayesian perspective the assumption of normal noise is incomprehensible since it does not constitute a reasonable generative model of the data. However, the method often works surprisingly well in practice. Table 6.2 presents error rates obtained by label regression on the previously described benchmark data sets, again using the isotropic squared exponential covariance function (6.11). It can be observed that the average error rate obtained by label regression is worse than for the probit model on the Ionosphere and Wisconsin data sets, while for the Pima Indians and the Crabs data sets the performance is similar. But for the (higher dimensional) Sonar problem we observe a lower error rate and especially for the USPS digits it is almost a surprising reduction of 50% compared to the probit model with EP approximation.

From a Bayesian point of view these results are surprising since the model—as a generative model—seems by far less appropriate for the binary classification task. The phenomenon could be explained as an effect of severe model mismatch, especially for the USPS data. For this kind of data, using a covariance function based on pixel-wise squared distance is probably very inappropriate for describing the similarity between images. Moreover, using the squared exponential covariance function corresponds to a strong prior belief about the smoothness of the latent function. Thereby it is reasonable to assume that the GPC model as well as the label regression approach are both unsuited for describing the data generating process, but label regression accidentally happens to work better.

A more technical insight can be found by relating label regression to Fisher's (1936) linear discriminant analysis via minimum squared error procedures as described by Duda and Hart (1973, ch. 5.8.2). Although leaving the Bayesian framework, these methods

Table 6.2.: Further error rates for benchmark data sets. The first three columns give the name of the data set, number of observation m and dimension of inputs n. All error rates are estimates obtained by 10 fold cross-validation. The error rates for Laplace's approximation, EP and Support Vector Machines are taken from Table 6.1. Additionally the results of k-nearest neighbour classification (kNN) are shown, where the number of neighbours is found using cross-validation on the training set respectively. The last column shows the error rates obtained by label regression (LR).

			Laplace	EP	SVM	kNN	LR
Data Set	m	n	E	E	E	E	E
Ionosphere	351	34	8.84%	7.99%	**5.69%**	12.63%	10.54%
Wisconsin	683	9	**3.21%**	**3.21%**	**3.21%**	**3.21%**	4.23%
Pima Indians	768	8	22.77%	**22.63%**	23.01%	24.69%	22.87%
Crabs	200	7	**2.0%**	**2.0%**	**2.0%**	5.0%	**2.0%**
Sonar	208	60	15.36%	13.85%	**11.14%**	13.93%	11.29%
USPS 3's vs. 5's	1540	256	2.27%	2.21%	2.01%	3.12%	**1.23%**

have a geometrical interpretation in the weight-space view, see Section 3.4.1. In the context of kernel machines this approach is also known as kernel Fisher discriminant analysis (Mika et al., 1999), least-squares Support Vector Machines (Suykens and Vandewalle, 1999), and kernel ridge regression (Saunders et al., 1998). Note that the posterior mean function in a Gaussian process regression model with normal noise (3.12a) has exactly the same form as the kernel ridge regression estimate, which is derived from ridge regression as proposed by Hoerl and Kennard (1970) by using the kernel trick. For a detailed description of the relations between these methods see Kuss (2002, ch. 2).

6.4. Conclusions & Discussion

The presented experiments reveal serious differences between Laplace's method and Expectation Propagation when used in the Gaussian process model for binary classification. The results corroborate the considerations about the two approximations based on the structure of the posterior given in Section 6.2. Although only a handful of data sets have been used in the study, we believe the conclusions to be well-founded and generally valid.

From the structural properties of the posterior we described why Laplace's method systematically underestimates the mean $\mathbf{m}$. The resulting approximate posterior $f|\mathcal{D}$ over latent functions will have too small amplitude, although the sign of the mean function will be mostly correct. As an effect Laplace's method gives over-conservative predictive probabilities, and diminished information about the test labels. This effect has been shown empirically on several real world examples. Large resulting discrepancies in the actual posterior probabilities were found, even at the training locations, which renders the predictive class probabilities produced under this approximation grossly inaccurate. Note, the difference becomes less dramatic if we only consider the classification error rates obtained by thresholding p^* at 1/2. For this particular task, we have seen that the sign of the latent function tends to be correct (at least at the training locations). However, the performance on benchmark data sets also revealed the error rates obtained by Laplace's method to be inferior to EP results.

The EP approximation has shown to give results very close to MCMC both in terms of predictive distributions and evidence estimates. We have shown and explained why the marginal distributions of the posterior can be well approximated by Gaussians.

Further, the evidence values obtained by Laplace's method and EP differ systematically which will lead to different results of ML-II hyper-parameter estimation. The discrepancies are similar for different tasks. We were able to exemplify that the EP approximation of the evidence is accurate. To show this we used Annealed Importance Sampling to obtain unbiased estimates of the evidence for Gaussian process models.

For certain data sets numerical problems occurred when using the EP approximation for ML-II hyper-parameter estimation because the signal variance σ_s^2 can become extremely large. The problem stems from the property that for large values of σ_s^2 the evidence becomes very insensitive to changes in σ_s^2 and so the optimisation algorithm may evaluate the approximate evidence for extreme values of σ_s^2. At this point it is recommended to take another look at Figure 6.3(b). Intuitively, for large signal variances the prior becomes more spread out such that the likelihood becomes more and more similar to a hard truncation. The evidence equals the probability mass of the prior in the orthant that is left after truncation. But the probability mass in any of the orthants remains constant if only the signal variance is changed for fixed correlation structure. The previous argument was based on the assumption that the likelihood implements a hard truncation, which is only an approximation, but this approximation becomes better the larger σ_s^2 is. Note that this insensitivity of the evidence with respect to changes in the signal variance can already be observed in the upper parts of Figures 6.6(2a) and 6.8(2a). A possible solution to this problem is to introduce a hyper-prior on σ_s^2 which ensures that the parameter stays in a reasonable range.

In the experiments above a stationary covariance function (6.11) was used. This corresponds to the belief that the latent function has stationary characteristics over the whole input space. For certain data this might not be one's true prior belief. If the classes are well separated, one might expect the latent function f to be relatively constant over the support of $p(\mathbf{x}|y)$ of the respective classes and to make a sharp transition at the decision boundary. Another property of the isotropic zero-mean prior is that the predictive probability goes towards 1/2 in parts of the input space where no observations have been made. This behaviour could be controlled by introducing a non-zero mean function, for example by adding a linear term to the covariance function.

In the experiments summarised in Table 6.1 we compared the predictive accuracy of GPC to Support Vector Machines. While the SVMs show a tendency to give lower error rates, the information contained in predictive distributions seems comparable. An obvious question is, why the SVMs often give lower error rates than Gaussian process classification? Unfortunately we are not able to identify a single explanation for this. One aspect of the answer might be that the Support Vector Machine directly aims at minimising the error rate, while the GPC model aims at giving a full predictive distribution. Moreover, in the experiments we used a soft-margin variant of the SVM, which tolerates the occasional misclassification of training examples, so called margin errors. Thereby the soft-margin SVM solution is less sensitive to outliers—either outliers in $\mathbf{x}$ or erroneous labels y—which could not be ignored in the GPC model. Another aspect that complicates the interpretation of the results is that we used different strategies for parameter estimation, i.e. model selection. Conceptually GPC comes with the advantage that Bayesian model selection can be used to set hyper-parameters by ML-II estimation, while the parameters of an SVM usually have to be set by cross-validation (gradient based methods exist, e.g. Chapelle et al. (2002)). We have not exploited this feature in the above comparison, but the ability to handle multiple free parameters should enable practitioners to use more complex covariance functions, that are designed for particular applications. However, as described in Section 2.3.2, the maximum likelihood II approach can only be expected to work well, the more the model is capable of approximating the data generating process. Otherwise ML-II estimation will give parameters that make the data to appear as expected as possible, but the predictive performance can be poor, see again Figure 5.1 (on page 58) for an example in the regression context. Cross validation is less sensitive to this form of model mismatch, since it directly optimises an approximate measure of the predictive performance we are interested in. In fact, it would be interesting to use cross-validation for the GPC model as well. If cross-validation shows to work better than ML-II estimation, this could be a strong indication for model mismatch. It may also be insightful to test whether other covariance functions could be used to reduce the model mismatch, e.g. by comparing the corresponding evidences and predictive performances.

As mentioned above, in the GPC model we did not consider the possibility that some observations are mislabelled. The likelihood can be modified such that label errors are explicitly taken into account, giving a robust classification model, see Copas (1988) or Wood and Kohn (1998) for related approaches. Let π denote the probability by which an observation has an erroneous label. In this case the likelihood becomes a mixture

$$\begin{aligned} p(y|f(\mathbf{x}),\pi) &= (1-\pi)\,p(y|f(\mathbf{x})) + \pi\,(1-p(y|f(\mathbf{x}))) && (6.18\text{a}) \\ &= \pi + (1-2\pi)\,p(y|f(\mathbf{x})) && (6.18\text{b}) \end{aligned}$$

which can be seen as another instance of a "two-model-model" as described in Section 5.3 for robust regression. If $p(y|f)$ is chosen to be the probit model $\Phi(fy)$, inference can be approximated using Expectation Propagation, for which the necessary moments can be derived from eqs. (6.10). Note that the mixture likelihood is not log-concave and therefore the posterior can be multimodal. Minka (2001a) describes EP for classification with label errors for the Bayes Point Machine of Herbrich et al. (2001), which is similar to this model.

An extension to the classification setting is the semi-supervised scenario, in which additional unlabelled inputs $\mathbf{X}_{UL}$ are available. Semi-supervised variants of Gaussian process classification can be constructed by incorporating $\mathbf{X}_{UL}$ into the likelihood

$$p(\mathbf{y},\mathbf{X}_{UL}|f,\mathbf{X},\boldsymbol{\theta}) \;=\; p(\mathbf{y},|\mathbf{f},\boldsymbol{\theta})\,p(\mathbf{X}_{UL},|\mathbf{f}_{UL},\boldsymbol{\theta})\,, \qquad (6.19)$$

where $p(\mathbf{X}_{UL}, |\mathbf{f}_{UL}, \boldsymbol{\theta})$ can be chosen according to the assumed relation between the latent function and the unlabelled observations. Lawrence and Jordan (2005) propose such a model, by which a latent function f is more likely, if its value is away from zero at the unlabelled inputs. Intuitively, this makes the latent function switch sign in areas where no unlabelled inputs are available, if not encouraged by the labelled observations. In other words, the latent function is encouraged to have a strong opinion on each unlabelled observation, additionally to explaining the labelled instances. In general, semi-supervised learning will often lead to multimodal posteriors, each mode corresponding to a specific assignment of the unlabelled observations into classes, similar to the posterior in the robust regression model with mixture noise.

In summary, we found that EP is the method of choice for approximate inference in binary GPC models, when the computational cost of MCMC is prohibitive. Very good agreement is achieved for both predictive probabilities and evidence estimates. In contrast, the Laplace approximation is so inaccurate that we advise against its use, especially when predictive probabilities are to be taken seriously.

6.5. Bibliographical Remarks

The Gaussian process model for binary classification can be seen as a non-parametric treatment of Bayesian logistic regression as described by Albert and Chib (1993). The probit likelihood is also common in frequentist generalised linear models, see e.g. McCullagh and Nelder (1989) or Collet (1991).

Gaussian process classification has been described by Williams and Barber (1998) using Laplace's approximation while Neal (1998a) proposed the use of Hybrid Monte Carlo sampling. Sparse EP approximations for the classification model have been proposed for example by Csató and Opper (2002), Seeger (2002), and Lawrence et al. (2003). Seeger (2002) also described how the PAC-Bayesian framework of McAllester (1999) can be applied to Gaussian process classification, giving highly non-trivial error bounds.

In this chapter we only considered binary classification. For polychotomous problems a common strategy is to decompose the problem into several one-against-all or one-against-one classification problems, which can be handled using the binary classification model. Williams and Barber (1998) also describe a multi-class GP model in which several latent functions are combined using a softmax type likelihood. Inference can be approximated using Laplace's method (Williams and Barber, 1998), (sparse) EP approximations (Seeger and Jordan, 2004), or variational methods (Girolami and Rogers, 2005). However, it is not clear which approach to multi-class problems works best in practice, see e.g. Rifkin and Klautau (2004).

As discussed above model selection by maximum likelihood II estimation is based on the assumption that the assumed model matches the underlying data generating process well. Furthermore, the approach is prone to over-fitting especially if the number of hyper-parameters is large. Based on the EP approximation Qi et al. (2004) describe an alternative predictive approach for the Gaussian process model for classification, similar to framework of Geisser and Eddy (1979) as described in Section 3.2.1 for regression.

Above the performance of GPC was compared to Support Vector Machines, which at present is probably the most successful and popular classification algorithm. The SVM origins in the work of Vapnik on statistical learning theory, see Vapnik (1998, 1999) and Schölkopf and Smola (2002, ch. 7) for further details. Several attempts have been made to link Support Vector Machines to Gaussian process classification models, e.g. Seeger (2000) and Sollich (2002). Sollich (2002) constructs a prior and a likelihood such that the Support Vector Machine is equivalent to a maximum *a posteriori* estimate in this model, which is not exactly a GP model as described in Chapter 3. The problem is that the hinge loss, as used in Support Vector Machines, cannot be written as the negative log of a sampling distribution $p(y|f)$.

7. Gaussian Processes for Reinforcement Learning

As soon as we admit the existence of a lack of completely accurate measurement, or, as we shall say, uncertainty, in one part of the system, we must allow for it in every part. — Bellman (1967, p. 7)

Reinforcement learning refers to the problem of how an agent can maximise its utility in a sequential decision problem. The crucial question is how the agent should adapt its behaviour as a result of its experience. An important aspect is the evaluation of alternative actions in the light of uncertainty about future consequences of decisions. The problem has been studied in various contexts, for example in statistical decision theory, control engineering, artificial intelligence, and operations research. This chapter presents some ideas for applications of Gaussian process regression models for reinforcement learning in continuous state spaces and discrete time.

7.1. Reinforcement Learning

In the beginning some elementary concepts of reinforcement learning will be described, which constitute the basis for the subsequent developments. The generic setting is that an agent interacts with an environment in a sequence of actions and receives corresponding reward signals. The aim of the agent is to optimise its strategy, i.e. the choice of action in a given situation, in order to maximise a measure of utility. A formally simple and yet powerful framework to represent the agent's environment is the discrete-time Markov decision process (MDP) as will be described in Section 7.1.1.

Characteristic to the problems studied in reinforcement learning is that actions can have long-term effects in later stages of the decision process. Therefore, cause and (delayed) effect are difficult to relate, which is known as the temporal credit assignment problem. The greedy myopic strategy to always choose the action that promises maximum immediate reward is typically sub-optimal and strategic planning of future actions is crucial. This problem is explicitly recognised by the distinction between short-term (reward) and long-term (value) desiderata. In contrast to the immediate reward, the value describes the strategic validation of a situation. The consistency between rewards and values are expressed by the Bellman equation. Both value functions and the Bellman equation will be described in Section 7.1.2.

If the environment has finitely many states and the characteristics of the decision process are known for certain, the optimal strategy and the corresponding value function can be computed using dynamic programming techniques (Section 7.1.3). However, the more challenging and relevant question is how to act if the agent is uncertain about the consequences of its actions. Temporal difference learning, as will be briefly described in Section 7.1.4, refers to techniques that can be used to evaluate strategies while interacting with an uncertain environment. The problems becomes more difficult if the decision problems involve continuous variables. In this case function approximation techniques have to be used, as will be described in Section 7.3.

In order to make decisions that anticipate future consequences, the agent needs a model of the environment. Depending on the structure of the environment this model can be of various forms. A key idea in later sections will be that Gaussian process regression models can be used to describe the agent's uncertainty in certain kinds of environments. Having a probabilistic model allows the implementation of strategic planing of future actions. Furthermore, using a GP to represent the value function, the Bellman equations can be approximated in continuous MDPs (Section 7.3.1).

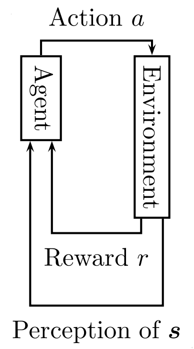


Figure 7.1.: The reinforcement learning setting.

7.1.1. Markov Decision Processes

Let $s_t \in \mathcal{S}$ denote the state of the decision process at time t where $t = 0, 1, 2, \ldots$ indexes discrete stages and $\mathcal{S}$ denotes the set of all possible states. The state is a description of the environment at a particular time, which, assuming that the state is fully observable, captures all information available to the agent in order to make a decision on the action. In each stage t the agent selects exactly one action $a_t \in \mathcal{A}(s_t)$ from a set of feasible actions $\mathcal{A}(s_t)$ in the current state.

An action results in a state transition to a potentially different consecutive state s'. The dynamics of the environment are described in terms of state transition probabilities $p(s'|s, a)$ which define a distribution over consecutive states $s' \in \mathcal{S}$ for all possible combinations of states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}(s)$. A sequential decision process is Markovian if the state transition probabilities only depend on the current state and action, independent of the history of the decision process. Below it will always be assumed that the environment is stationary, i.e. the state transition probabilities do not change over time. If the sets of possible states $\mathcal{S}$ and actions $\mathcal{A}(s)$ are finite, the sequential decision process is called a finite MDP. For finite MDPs the state transition probabilities are multinomial distributions which can be represented in the form of tables.

With each state transition (s, a, s') the agent immediately receives a reward signal $r \in \mathbb{R}$. The reward can be stochastic $p(r|s, a, s')$ and the expected reward is denoted by

$$\mathsf{r}(s, a, s') = \mathbb{E}\left[r|s, a, s'\right] \tag{7.1}$$

which is assumed to be bounded.

A strategy or policy $\mathfrak{p}$ specifies a distribution over actions $p_{\mathfrak{p}}(a|s)$ for each $s \in \mathcal{S}$ where $a \in \mathcal{A}(s)$ respectively. A policy is called pure or deterministic if in each state a particular action $\mathfrak{p}(s) \to a$ is selected with probability one, otherwise the policy is called stochastic. If a policy is independent of the stage t of the decision process, it is referred to as a stationary policy.

One also distinguishes according to the length, i.e. the number of stages, of a decision process between tasks of finite and infinite length. If a process terminates after a finite but unknown number of stages, this is also called an indefinite MDP.

7.1.2. Value Functions and Bellman Equations

As the decision process advances, the agent receives a sequence of reward signals. The agent's objective is to act as to maximise its utility. The utility characterises the valuation of rewards over time and so expresses the agent's inter-temporal preferences.

Various forms of utility functions can be found in the literature. For MDPs of finite length any function monotonically increasing in the sum of rewards is a legitimate choice. Let N denote the finite length of a decision process, the agent might be interested in maximising the weighted sum of rewards

$$\sum_{t=0}^{N} \gamma^t r_{t+1} \tag{7.2}$$

where $\gamma > 0$ is a parameter describing inter-temporal preferences, which is also referred to as *discount rate*. When assuming a utility function of this form in MDPs of infinite length, N has to be interpreted as the time-horizon of the agent, i.e. the number of future rewards that are relevant for the agent.

For tasks of potentially infinite length with an infinite horizon it must be assured that the utility is finite. The most common model is the geometrically discounted sum

$$\sum_{t=0}^{\infty} \gamma^t r_{t+1} \tag{7.3}$$

where the discount rate $\gamma \in [0, 1)$ ensures finiteness. The smaller the discount rate γ the more the agent prefers instant reward relative to rewards in the future. Although discounting is natural in many economic decision problems, in other domains it is often difficult to justify this particular form of utility function. Another value function which can be used in MDPs of infinite length is the average reward, see for example Arapostathis et al. (1993). However, the infinite geometrically discounted sum (7.3) is the most common utility function in the literature, because of its convenient analytical properties.

Many of the problems analysed in reinforcement learning—and obviously in real life—are actually decision problems of finite length. Decision problems of finite length are more difficult in the sense that the optimal action for a given state can depend on the remaining number of stages and therefore the optimal policy can be non-stationary. Hence, the infinite discounted model (7.3) is often used for analytical convenience because the optimal policy is stationary. Often the infinite length MDP model must be interpreted as an approximation to a finite but long decision process. A common remedy is to introduce an absorbing goal state which can be reached in finite time and in which the agent remains afterwards.

From the utility of an infinite discounted sequence of rewards (7.3) we now derive the notion of the value of a state under a given policy. The value of a state $\boldsymbol{s}$ under a given policy $\mathfrak{p}$ is defined as the sum of expected discounted future rewards starting in state $\boldsymbol{s}$:

$$\mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}) \;=\; \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | \boldsymbol{s}_0 = \boldsymbol{s}\right] \tag{7.4}$$

where the expectation is over future state transitions and associated rewards when following $\mathfrak{p}$. We refer to $\mathsf{V}^{\mathfrak{p}} : \mathcal{S} \to \mathbb{R}$ as the *state-value* function for policy $\mathfrak{p}$ in an MDP of infinite length.

Value functions define a partial ordering on the set of policies. A policy $\mathfrak{p}_A$ is weakly dominated by another policy $\mathfrak{p}_B$ if $\mathsf{V}^{\mathfrak{p}_A}(\boldsymbol{s}) \leq \mathsf{V}^{\mathfrak{p}_B}(\boldsymbol{s})$ for all $\boldsymbol{s} \in \mathcal{S}$.[1] Using this ordering we can characterise an optimal policy $\mathfrak{p}^\star$ by

$$\mathsf{V}^{\mathfrak{p}^\star}(\boldsymbol{s}) \;\geq\; \mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}) \quad \forall\, \boldsymbol{s} \in \mathcal{S}, \forall\, \mathfrak{p} \in \mathfrak{P} \tag{7.5}$$

where $\mathfrak{P}$ denotes all possible policies. Note that the optimal policy $\mathfrak{p}^\star$ is not necessarily unique. However, the unique optimal value function corresponding to all optimal policies will be denoted by $\mathsf{V}^\star$.

Due to the Markov assumption the value function V can be written as a recursive relation between states

$$\begin{aligned} \mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | \boldsymbol{s}_0 = \boldsymbol{s}\right] = \mathbb{E}\left[r_1 + \sum_{t=1}^{\infty} \gamma^t r_{t+1} | \boldsymbol{s}_0 = \boldsymbol{s}\right] && \text{(7.6a)} \\ &= \sum_{a \in \mathcal{A}(\boldsymbol{s})} p_{\mathfrak{p}}(a|\boldsymbol{s}) \sum_{\boldsymbol{s}' \in \mathcal{S}} p(\boldsymbol{s}'|\boldsymbol{s}, a) \left[\mathsf{r}(\boldsymbol{s}, a, \boldsymbol{s}') + \gamma \mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}')\right] && \text{(7.6b)} \end{aligned}$$

which gives the Bellman equation (7.6b) for the state-value function (Bellman, 1972). The Bellman equation reveals that the value of a state $\boldsymbol{s}$ under a fixed policy $\mathfrak{p}$ equals the expected immediate reward plus the discounted expected value of the consecutive state. Furthermore, the Bellman equations for all states ensure consistency between values of all states under a given policy $\mathfrak{p}$. Watkins (1989) introduced the *state-action-value* function

$$\begin{aligned} \mathcal{Q}^{\mathfrak{p}}(\boldsymbol{s}, a) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | \boldsymbol{s}_0 = \boldsymbol{s}, a_0 = a\right] && \text{(7.7a)} \\ &= \sum_{\boldsymbol{s}' \in \mathcal{S}} p(\boldsymbol{s}'|\boldsymbol{s}, a) \left[\mathsf{r}(\boldsymbol{s}, a, \boldsymbol{s}') + \gamma\, \mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}')\right] && \text{(7.7b)} \end{aligned}$$

which is the discounted sum of expected rewards when choosing action a in state $\boldsymbol{s}$ and following $\mathfrak{p}$ afterwards. The state-value function and the state-action-value function are

[1] If for all $\boldsymbol{s} \in \mathcal{S}$ the inequality $\mathsf{V}^{\mathfrak{p}_A}(\boldsymbol{s}) < \mathsf{V}^{\mathfrak{p}_B}(\boldsymbol{s})$ holds, the policy $\mathfrak{p}_B$ is said to be *strictly* dominant and *weakly* dominant if only the non-strict $\leq$ holds.

related by

$$\mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}) = \sum_{a \in \mathcal{A}(\boldsymbol{s})} p_{\mathfrak{p}}(a|\boldsymbol{s})\, \mathcal{Q}^{\mathfrak{p}}(\boldsymbol{s}, a) \; . \tag{7.8}$$

In finite Markov decision processes the values of all states can be stored in a table. In case the number of states becomes too large and especially for continuous $\mathcal{S}$, function approximation techniques have to be used, as will be described in Section 7.3.

7.1.3. Policy Iteration in Finite Markov Decision Processes

The problem of optimal sequential decision making is to find a policy that maximises the expected utility. In case of finite $\mathcal{S}$ and perfect information about the state transition probabilities and the reward structure, dynamic programming techniques can be used to compute the optimal value function $\mathsf{V}^{\star}$ and so the optimal policy $\mathfrak{p}^{\star}$ for all states.

In general we refer to the process of finding the value function corresponding to a given policy as *policy evaluation*. For finite Markov decision processes and given policy $\mathfrak{p}$ we can compute the probabilities

$$p(\boldsymbol{s}'|\boldsymbol{s}) \;=\; \sum_{a \in \mathcal{A}(\boldsymbol{s})} p(\boldsymbol{s}'|a, \boldsymbol{s})\, p_{\mathfrak{p}}(a|\boldsymbol{s}) \tag{7.9}$$

by averaging over the probabilities of taking actions according to the policy $\mathfrak{p}$. Note that $p(\boldsymbol{s}'|\boldsymbol{s})$ is a multinomial distribution for each $\boldsymbol{s} \in \mathcal{S}$. So we can define the $[|\mathcal{S}| \times |\mathcal{S}|]$ state transition probability matrix $\mathbf{T}$ element-wise:

$$T_{ij} \;=\; p(\boldsymbol{s}' = \boldsymbol{s}_i | \boldsymbol{s} = \boldsymbol{s}_j) \; , \tag{7.10}$$

which gives a stochastic matrix, sometimes also called the Markov matrix, see for example White (1993, ch. 1).

At first, it will be described how we can compute the $[|\mathcal{S}| \times 1]$ vector of values $\mathbf{v}$, such that $v_i = \mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}_i)$. The Bellman equations (7.6b) for all $|\mathcal{S}|$ states constitute a system of linear equations with $\mathbf{v}$ as a fixed point, as will be shown below.

Let $\mathbf{r}$ be the vector of expected rewards such that the ith entry corresponds to the expected reward of state transitions starting in $\boldsymbol{s}_i$ averaged over the actions selected by $\mathfrak{p}$ and the corresponding state transition probabilities

$$r_i \;=\; \sum_{a \in \mathcal{A}(\boldsymbol{s}_i)} \sum_{\boldsymbol{s}' \in \mathcal{S}} \mathsf{r}(\boldsymbol{s}_i, a, \boldsymbol{s}')\, p_{\mathfrak{p}}(a|\boldsymbol{s}_i) \; , \tag{7.11}$$

where $\mathsf{r}(\boldsymbol{s}_i, a, \boldsymbol{s}')$ is given by eq. (7.1). So we can write the $|\mathcal{S}|$ Bellman equations in matrix form

$$\mathbf{v} = \mathbf{r} + \gamma\, \mathbf{T}\mathbf{v} \tag{7.12}$$

and solve the system $(\mathbf{I} - \gamma\, \mathbf{T})\, \mathbf{v} = \mathbf{r}$ for $\mathbf{v}$. This can either be done directly $\mathbf{v} = (\mathbf{I} - \gamma\, \mathbf{T})^{-1}\, \mathbf{r}$ or by using fixed point iteration.

In order to use a fixed point iteration scheme we have to show that the Bellman equations (7.12) constitute a contraction mapping, i.e. the mapping has a Lipschitz constant smaller than one. Let $B(\mathbf{v}) = \mathbf{r} + \gamma\, \mathbf{T}\mathbf{v}$ be a self-mapping $B : \mathbb{R}^{|\mathcal{S}|} \rightarrow \mathbb{R}^{|\mathcal{S}|}$ then we have to show that $\|B(\mathbf{v}_1) - B(\mathbf{v}_2)\| = \lambda \|\mathbf{v}_1 - \mathbf{v}_2\|$ with $\lambda < 1$ for all $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^{|\mathcal{S}|}$, where $\|\cdot\|$ denotes the maximum norm in this case. By simply substituting the definition of B we obtain

$$\|B(\mathbf{v}_1) - B(\mathbf{v}_2)\| \;=\; \|\mathbf{r} + \gamma\mathbf{T}\mathbf{v}_1 - \mathbf{r} - \gamma\mathbf{T}\mathbf{v}_2\| \tag{7.13a}$$

$$\;=\; \|\gamma\mathbf{T}(\mathbf{v}_1 - \mathbf{v}_2)\| \;=\; \gamma\, \|\mathbf{T}\|\, \|\mathbf{v}_1 - \mathbf{v}_2\| \tag{7.13b}$$

and since $\gamma \in [0, 1)$ by definition and $\|\mathbf{T}\| \leq 1$ as a standard result for stochastic matrices, we observe that B is contracting. So we can implement policy evaluation by iteratively using $\mathbf{v}^{\text{new}} \leftarrow \mathbf{r} + \gamma\mathbf{T}\mathbf{v}^{\text{old}}$ as an update rule which is guaranteed to converge.

Once the values of states under a policy are found, we can ask the question how the policy might be modified to increase the value of each state and thereby to improve the policy. This process is referred to as a *policy improvement* step. Given $\mathbf{v}$ for a policy $\mathfrak{p}$ we obtain a $\mathfrak{p}^+$ which weakly dominates $\mathfrak{p}$ by acting greedily with respect to $\mathbf{v}$. The policy $\mathfrak{p}^+$ is a deterministic policy which in a given state $\boldsymbol{s}$ chooses the action a which maximises the expected value:

$$\mathfrak{p}^+(\boldsymbol{s}) \;=\; \underset{a \in \mathcal{A}(\boldsymbol{s})}{\operatorname{argmax}} \sum_{\boldsymbol{s}' \in \mathcal{S}} p(\boldsymbol{s}'|\boldsymbol{s}, a)\, \left[\mathsf{r}(\boldsymbol{s}, a, \boldsymbol{s}') + \gamma \mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}')\right] \tag{7.14}$$

where ties can be broken arbitrarily. By construction $\mathsf{V}^{\mathfrak{p}^+} \geq \mathsf{V}^{\mathfrak{p}}$ with equality only for $\mathsf{V}^{\star}$, which is known as the *policy improvement* theorem. Note that as a consequence of the policy improvement theorem, there always exists an optimal policy which is deterministic by acting greedily w.r.t. the optimal value function.

We observe a certain duality between policies and value functions. For a given policy we can use the *policy evaluation* scheme to find the corresponding values for all states, and for a given value function we can use *policy improvement* to compute a weakly dominant policy. Note that the optimal policy and the optimal value function map on each other.

We can now state the *policy iteration* scheme which allows us to approach the optimal policy sequentially. Policy iteration starts from an arbitrary policy $\mathfrak{p}^0$ and alternates between *policy evaluation* and *policy improvement* until convergence to the optimal policy and corresponding value function (see Algorithm 3). Since in each policy improvement step the new policy is guaranteed to weakly dominate the previous policy, the scheme converges to an optimal policy, which is the greedy policy with respect to the optimal value function.

7.1.4. Temporal Difference Learning

We saw that perfect information about a (finite) MDP enables us to compute the optimal policy and the corresponding value function. However, a more relevant and interesting scenario is that the agent is uncertain about the characteristics of the MDP, i.e. $p(\boldsymbol{s}'|\boldsymbol{s}, a)$ and $p(r|\boldsymbol{s}, a, \boldsymbol{s}')$, such that exploration of the environment becomes an integral part of

Algorithm 3 Policy iteration scheme for finite MDPs

```
Given: $\gamma$, Initial policy $\mathfrak{p}^0$, state transition probabilities, distribution of rewards,
convergence threshold $\epsilon$
Initialise $\mathbf{v}$ arbitrarily (e.g. $\mathbf{v} \leftarrow \mathbf{0}$)
repeat
  1. Policy Evaluation
  Compute $\mathbf{r}$ and $\mathbf{T}$ from $\mathfrak{p}$
  repeat
    $\mathbf{v}^{\text{old}} \leftarrow \mathbf{v}$
    $\mathbf{v} \leftarrow \mathbf{r} + \gamma\,\mathbf{T}\mathbf{v}$
  until $\|\mathbf{v}^{\text{old}} - \mathbf{v}\| < \epsilon$
  2. Policy Improvement
  for all $\boldsymbol{s} \in \mathcal{S}$ do
    $\mathfrak{p}(\boldsymbol{s}) \leftarrow \operatorname*{argmax}_{a \in \mathcal{A}(\boldsymbol{s})} \sum_{\boldsymbol{s}' \in \mathcal{S}} p(\boldsymbol{s}'|\boldsymbol{s}, a)\,[\mathsf{r}(\boldsymbol{s}, a, \boldsymbol{s}') + \gamma\,\mathsf{V}(\boldsymbol{s}')]$
  end for
until $\mathsf{V}$ and $\mathfrak{p}$ converge to optimal $\mathsf{V}^\star$ and $\mathfrak{p}^\star$
```

finding an optimal policy. We will now turn to the questions how the value function of a fixed policy can be evaluated under these circumstances and furthermore how the agent can learn to improve its policy?

Sutton (1988) describes how temporal difference learning can be used to estimate the value $\mathsf{V}^{\mathfrak{p}}(\boldsymbol{s})$ for all $\boldsymbol{s} \in \mathcal{S}$ under a fixed policy $\mathfrak{p}$. The basic idea of temporal difference (TD) learning is to adjust estimates of values if expectations and observations differ. Assume the agent observes the state transition $(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$ and corresponding reward r_{t+1}.[2] In temporal difference learning this observation is used to update the current estimate of $\mathsf{V}(\boldsymbol{s}_t)$. The expectation of the quantity $r_{t+1} + \gamma\mathsf{V}(\boldsymbol{s}_{t+1})$ is by definition equal to $\mathsf{V}(\boldsymbol{s}_t)$ and if the observation disagrees with the agent's estimate of $\mathsf{V}(\boldsymbol{s}_t)$ the latter is adjusted towards the observation by

$$\mathsf{V}(\boldsymbol{s}_t) \leftarrow \mathsf{V}(\boldsymbol{s}_t) + \alpha_t \underbrace{[r_{t+1} + \gamma\mathsf{V}(\boldsymbol{s}_{t+1}) - \mathsf{V}(\boldsymbol{s}_t)]}_{\text{temporal difference error}} \tag{7.15}$$

where α_t is a learning rate. This update rule can be interpreted as a stochastic approximation as described by Robbins and Monro (1951). For stochastic approximation methods to converge the learning rate α_t has to decrease over time satisfying

$$\sum_{t=1}^{\infty} \alpha_t \;=\; \infty \qquad \text{and} \qquad \sum_{t=1}^{\infty} \alpha_t^2 \;<\; \infty \tag{7.16}$$

[2]Note that we index the observed reward of a state transition $(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$ by $t+1$, which corresponds to the notion that the reward is received in the consecutive state $\boldsymbol{s}_{t+1}$.

such as $\alpha_t = t^{-1}$. In the above update rule (7.15) the information contained in the observed state transition is used to update of the value of $\mathsf{V}(\boldsymbol{s}_t)$ only. The information can be passed back to the states previously visited by keeping eligibility traces which leads to the TD(λ) algorithm, see for example Sutton and Barto (1998, ch. 7).

Using the update rule (7.15) policy evaluation can be implemented, but it in general it is not obvious how to improve the policy if the state transition probabilities are unknown.

This problem can be overcome by using a state-action-value function $\mathcal{Q}$ as given by eq. (7.7). The greedy policy with respect to $\mathcal{Q}$ is to choose the action according to

$$\mathfrak{p}^{+}(\boldsymbol{s}) = \operatorname*{argmax}_{a \in \mathcal{A}(\boldsymbol{s})} \mathcal{Q}(\boldsymbol{s}, a) \tag{7.17}$$

where ties can be broken arbitrarily. However, always acting greedily would inhibit exploratory behaviour. Several heuristics have been proposed to balance exploration and exploitation, e.g. the ϵ-greedy policy is used which acts greedily with probability $(1-\epsilon)$ and chooses a random action with probability ϵ.

The state-action-value function (7.7) gave its name to $\mathcal{Q}$-Learning as described by Watkins and Dayan (1992). The objective in $\mathcal{Q}$-Learning is to use a state-action-value function to learn the optimal $\mathfrak{p}^{\star}$. Each observed state transition is used for an update

$$\mathcal{Q}(\boldsymbol{s}_t, a_t) \leftarrow \mathcal{Q}(\boldsymbol{s}_t, a_t) + \alpha_t \left[r_{t+1} + \gamma \max_{a \in \mathcal{A}(\boldsymbol{s}_{t+1})} \mathcal{Q}(\boldsymbol{s}_{t+1}, a) - \mathcal{Q}(\boldsymbol{s}_t, a_t) \right] \tag{7.18}$$

where the a_{t+1} is taken greedily. As an effect of taking the maximum in the TD update, $\mathcal{Q}$-Learning is an *off-policy* method, which means that it learns the optimal value function $\mathcal{Q}^{\star}$ even if a different policy $\mathfrak{p}$ is used by the agent to generate the observed state transitions (the *on-policy*).

In order for $\mathcal{Q}$-Learning to converge, the only requirement on the on-policy is that it should explore each state-action combination infinitely many times. Furthermore, the learning rate α_t has to decay as described by eq. (7.16) which is common to all stochastic approximation methods.

Algorithm 4 $\mathcal{Q}$-Learning for finite MDPs

```
Given: $\gamma$, learning rate scheme $\alpha_t$, initial state $\boldsymbol{s}_0$, on-policy $\mathfrak{p}$
Initialise: $\mathcal{Q}(\boldsymbol{s}, a) \leftarrow 0$ for all $\boldsymbol{s} \in \mathcal{S}$ and $a \in \mathcal{A}(\boldsymbol{s})$
for $t = 0, 1, 2, \ldots$ do
    Choose $a_t$ from on-policy $\mathfrak{p}$, e.g. $\epsilon$-greedy from $\mathcal{Q}(\boldsymbol{s}_t, a)$
    Observe state transition to $\boldsymbol{s}_{t+1}$ and reward $r_{t+1}$
    $\mathcal{Q}(\boldsymbol{s}_t, a_t) \leftarrow \mathcal{Q}(\boldsymbol{s}_t, a_t) + \alpha_t \left[ r_{t+1} + \gamma \max_{a \in \mathcal{A}(\boldsymbol{s}_{t+1})} \mathcal{Q}(\boldsymbol{s}_{t+1}, a) - \mathcal{Q}(\boldsymbol{s}_t, a_t) \right]$
end for
```

7.2. Model Identification and Propagation of Uncertainty

Temporal difference methods as described above aim at estimating value functions online but do not build an explicit model of the state transition probabilities $p(\boldsymbol{s}'|\boldsymbol{s}, a)$ and the distribution of rewards $p(r|\boldsymbol{s}, a, \boldsymbol{s}')$. Each observed state transition is used only once to update the current estimate of the value of a state but is forgotten afterwards. Note that by keeping eligibility traces, e.g. as in TD(λ) and $\mathcal{Q}(\lambda)$, the information contained in an observed state transition is also used to update the values of states visited previously. However, the state transition itself is not remembered, in particular it is not used to reduce the agent's uncertainty about the dynamics of its environment. Such methods are called *model free* or *direct* reinforcement learning. Model free approaches are computationally simple, but usually do not exploit all information contained in the observation relevant for optimal decision making.

Another approach is to build a model of the environment and the reward structure. Such a probabilistic model of the environment, also called *world* model, can be used to simulate consequences of actions and so planning can be implemented. Model based methods are also referred to as *indirect* reinforcement learning.

For instance in the Dyna architecture proposed by Sutton (1990, 1991) a *world model* of the state transition probabilities and the associated rewards is used. An observed state transition and the corresponding reward are used for a temporal difference update of the value function (either V or $\mathcal{Q}$) and for updating the world model. Between updates based on real experience, the world model is used to update the value function by simulated state transitions. This way the information gained by real experience is propagated and the value function is in part adjusted to agree with the world model. Implementing Dyna architectures for finite MDPs using table based representations is a simple extension of the methods described above. As to be expected, experiments show that far fewer real observations of the environment are necessary to find the optimal policy compared to model-free methods (Sutton and Barto, 1998, ch. 9).

From a Bayesian point of view reinforcement learning is a sequential decision problem in which each observed state transition reduces the agent's uncertainty about its environment. In each stage a Bayesian agent chooses an action given some notion of optimality, i.e. minimising some form of expected loss. An intuitively appealing criterion is to minimise the *regret*, i.e. the number of sub-optimal decisions. Note that this measure is meaningful in MDPs of both finite and infinite length. When trying to minimise the regret, the problem is that for choosing an action in an uncertain environment two desiderata have to be balanced. The agent can exploit its experience and choose the action which seems optimal in the present state of information. On the other hand, the agent could also explore the environment by choosing an action which has an uncertain effect. The observation could have greater novelty and could improve the quality of later decisions. However, how to find the optimal trade-off between *exploration* and *exploitation* which minimises the regret is an open problem for all but the simplest environments. To formalise a suitable loss function and computing its expectation over the uncertainty in the environment is an unsolved problem for general MDPs, see for example the early work by Martin (1967).

Compared to the above decision problem, implementing Bayesian inference about the environment is usually much simpler. For finite MDPs the state transition probabilities $p(\boldsymbol{s}'|\boldsymbol{s},a)$ are multinomial distributions and the conjugate multinomial-Dirichlet model can be used to describe the agent's uncertainties (Gelman et al., 1995, ch. 3.5). However, if the state space $\mathcal{S}$ is continuous, i.e. the representation of the state involves continuous variables, this approach becomes inapplicable. Instead, in this case we propose to use Gaussian process regression models to represent the agent's beliefs about the dynamics of its environment.

7.2.1. Prediction for Uncertain Inputs

In the following we will study the use of Gaussian process regression models for reinforcement learning. At various places we will make use of an approximation of the predictive distribution when the input itself is uncertain, as will be described in this—unfortunately rather technical—section. Approximations of this kind have been described previously, e.g. by Haylock and O'Hagan (1996), Girard et al. (2003), and Quiñonero-Candela et al. (2003).

In the conjugate Gaussian process regression model, as described in Section 3.2, the predictive distribution of the latent function value at a given test input $\mathbf{x}_*$ is univariate normal

$$p(f_*|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi},\mathbf{x}_*) \;=\; \mathcal{N}(f_*|m_*(\mathbf{x}_*),k_*(\mathbf{x}_*,\mathbf{x}_*)) \tag{7.19}$$

with mean and variance given by the mean and covariance function eqs. (3.12) of the posterior Gaussian process. As in Section 3.2, $\boldsymbol{\psi}$ denotes the hyper-parameters of the GP prior and $\boldsymbol{\theta}$ are the likelihood parameters, i.e. the noise variance σ_{n}^2 in this case. If the test input itself is a random variable following a multivariate normal distribution $\mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu},\boldsymbol{\Sigma})$, the predictive distribution averaged over the uncertainty in $\mathbf{x}_*$ is:

$$\begin{aligned} p(f_*|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\mu},\boldsymbol{\Sigma}) \;&=\; \int p(f_*|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi},\mathbf{x}_*)\,\mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu},\boldsymbol{\Sigma})\,d\mathbf{x}_* && \text{(7.20a)}\\ &\approx\; q(f_*|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\mu},\boldsymbol{\Sigma}) \;=\; \mathcal{N}(f_*|m_1,m_2) && \text{(7.20b)} \end{aligned}$$

which can be approximated analytically, see Figure 7.2 for an illustration. The approximation is found by computing the first and second moment of the predictive distribution (7.20a). In order to compute these moments analytically the prior covariance function is required to be a squared exponential (3.36) of the form

$$k(\mathbf{x},\mathbf{x}',\boldsymbol{\psi}) \;=\; \sigma_s^2 \exp\Big(-\tfrac{1}{2}\left(\mathbf{x}-\mathbf{x}'\right)^\top \mathbf{W}^{-1}\left(\mathbf{x}-\mathbf{x}'\right)\Big)\;, \tag{7.21}$$

or a polynomial covariance function (3.39), of which the latter will not be considered in the following. Note that mixtures of these covariance functions could be handled too.

The first moment of $p(f_*|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\mu},\boldsymbol{\Sigma})$ as given by eq. (7.20a) can be computed by

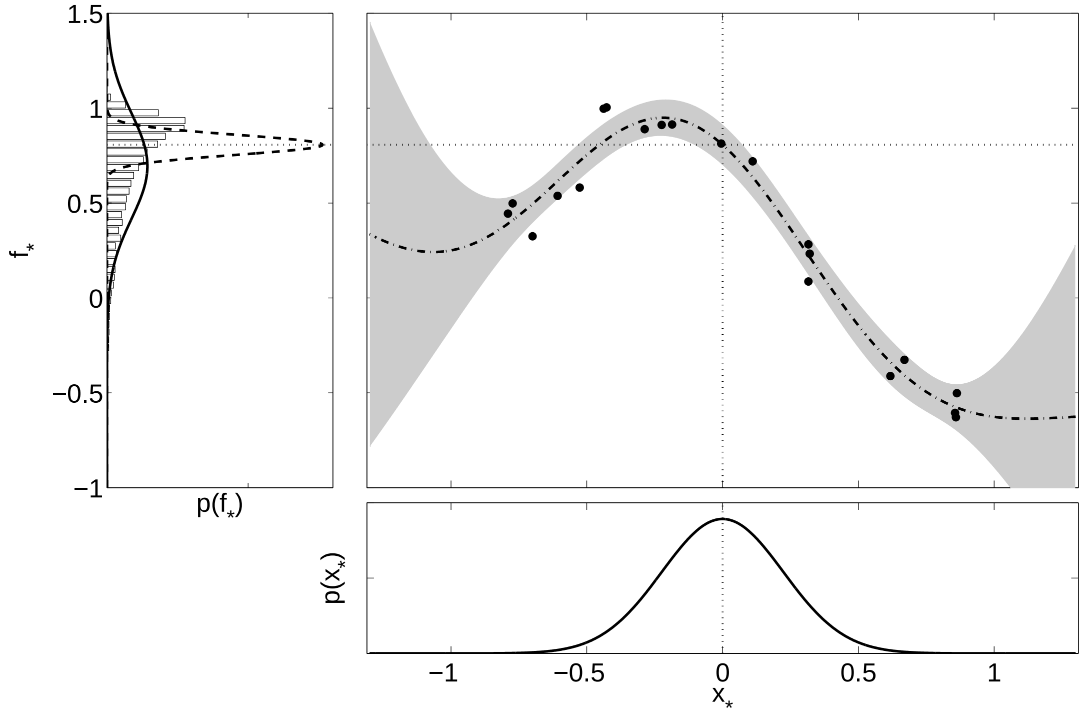


Figure 7.2.: Illustration of prediction for uncertain inputs in Gaussian process regression models with normal noise. The large panel shows a posterior Gaussian process inferred from the observations shown as points. The dashed line describes the mean function and the grey area depicts ±2 standard deviations. The lower panel shows the distribution of the uncertain input $\mathcal{N}(x_*|\mu_*, \sigma_*^2)$. From this distribution 10000 samples were generated and the corresponding predictive distributions of function values f_* were computed. The left hand panel shows a histogram of samples from these predictive distributions, i.e. samples of $p(f_*|\mathcal{D}, \boldsymbol{\psi}, \mu_*, \sigma_*^2)$, and the Gaussian approximation $\mathcal{N}(f_*|m_1, m_2)$, which matches the first two moments. The dashed line in the left panel illustrates the predictive distribution for a certain input $x_* = \mu_*$ (and $\sigma_*^2 = 0$).

taking the expectation

$$\begin{aligned} m_1 &= \int f_* \, p(f_*|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \, df_* && (7.22a) \\ &= \int \left[\int f_* \, p(f_*|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{x}_*) \, df_* \right] \mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \, d\mathbf{x}_* && (7.22b) \\ &= \int m_*(\mathbf{x}_*) \, \mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \, d\mathbf{x}_* && (7.22c) \end{aligned}$$

over the posterior mean function $m_*(\mathbf{x}_*)$ as given by (3.12a). Writing $m_*(\mathbf{x}_*) = \boldsymbol{\beta}^\top \mathbf{k}(\mathbf{x}_*)$ such that $\boldsymbol{\beta} = (\mathbf{K} + \sigma_{\mathrm{n}}^2 \mathbf{I})^{-1} \mathbf{y}$ and using the integral identities given in Appendix B.2.3

we get the expression:

$$\begin{aligned} m_1 &= \int (\boldsymbol{\beta}^\top \mathbf{k}(\mathbf{x}_*))\, \mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x}_* && (7.23a)\\ &= \sum_{i=1}^{m} \int \beta_i\, k(\mathbf{x}_*, \mathbf{x}_i)\, \mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x}_* && (7.23b)\\ &= \sum_{i=1}^{m} \beta_i\, \sigma_s^2 \left|\boldsymbol{\Sigma}\mathbf{W}^{-1} + \mathbf{I}\right|^{-\frac{1}{2}} \exp\Big(-\tfrac{1}{2}\,(\mathbf{x}_i - \boldsymbol{\mu})^\top (\boldsymbol{\Sigma} + \mathbf{W})^{-1} (\mathbf{x}_i - \boldsymbol{\mu})\Big) \end{aligned}$$

which can be written in the form $m_1 = \boldsymbol{\beta}^\top \mathbf{l}$. Intuitively the ith element of $\mathbf{l}$ is an expectation of $k(\mathbf{x}_i, \mathbf{x}_*)$ over $\mathbf{x}_*$.

The second central moment, i.e. the variance, of $p(f_*|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ as given by eq. (7.20a) can be decomposed into three expectations:

$$\begin{aligned} m_2 &= \int (f_* - m_1)^2\, p(f_*|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})\, df_* && (7.24a)\\ &= \mathbb{E}[m_*^2(\mathbf{x}_*)] + \mathbb{E}[\sigma_*^2(\mathbf{x}_*)] - \mathbb{E}[m_*(\mathbf{x}_*)]^2 && (7.24b) \end{aligned}$$

which will be computed separately. The first expectation results in a quadratic form

$$\mathbb{E}[m_*^2(\mathbf{x}_*)] \;=\; \int (\boldsymbol{\beta}^\top \mathbf{k}(\mathbf{x}_*))^2\, \mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x}_* \;=\; \boldsymbol{\beta}^\top \mathbf{L}\, \boldsymbol{\beta} \qquad (7.25)$$

where the elements of the $[m \times m]$ matrix $\mathbf{L}$ are given by

$$L_{ij} \;=\; \frac{k(\mathbf{x}_i, \boldsymbol{\mu}) k(\mathbf{x}_j, \boldsymbol{\mu})}{\sqrt{|2\boldsymbol{\Sigma}\mathbf{W}^{-1} + \mathbf{I}|}} \exp\Big((\mathbf{z}_{ij} - \boldsymbol{\mu})^\top \left(\tfrac{1}{2}\mathbf{W} + \boldsymbol{\Sigma}\right)^{-1} \boldsymbol{\Sigma}\mathbf{W}^{-1} (\mathbf{z}_{ij} - \boldsymbol{\mu})\Big) \qquad (7.26)$$

and $\mathbf{z}_{ij} = \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$ is an arithmetic average of the ith and jth input. The second expectation in eq. (7.24b) comes in the form of

$$\begin{aligned} \mathbb{E}[\sigma_*^2(\mathbf{x}_*)] &= \int \sigma_*^2(\mathbf{x}_*)\, \mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x}_* && (7.27a)\\ &= \sigma_s^2 - \int \mathbf{k}(\mathbf{x}_*)^\top (\mathbf{K} + \sigma_{\mathfrak{n}}^2 \mathbf{I})^{-1} \mathbf{k}(\mathbf{x}_*)\, \mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x}_* && (7.27b)\\ &= \sigma_s^2 - \operatorname{tr}\big((\mathbf{K} + \sigma_{\mathfrak{n}}^2 \mathbf{I})^{-1} \mathbf{L}\big) && (7.27c) \end{aligned}$$

which also depends on the matrix $\mathbf{L}$ as given by eq. (7.26). The third term in eq. (7.24b) is simply the squared value of the first moment (7.23). Putting everything together we get an analytic expression for the variance:

$$m_2 \;=\; \boldsymbol{\beta}^\top \mathbf{L}\, \boldsymbol{\beta} + \sigma_s^2 - \operatorname{tr}\big((\mathbf{K} + \sigma_{\mathfrak{n}}^2 \mathbf{I})^{-1} \mathbf{L}\big) - m_1^2\,. \qquad (7.28)$$

Furthermore, the predictive distributions of two independent Gaussian process models can become correlated for a random $\mathbf{x}_*$. Let A and B denote two Gaussian pro-

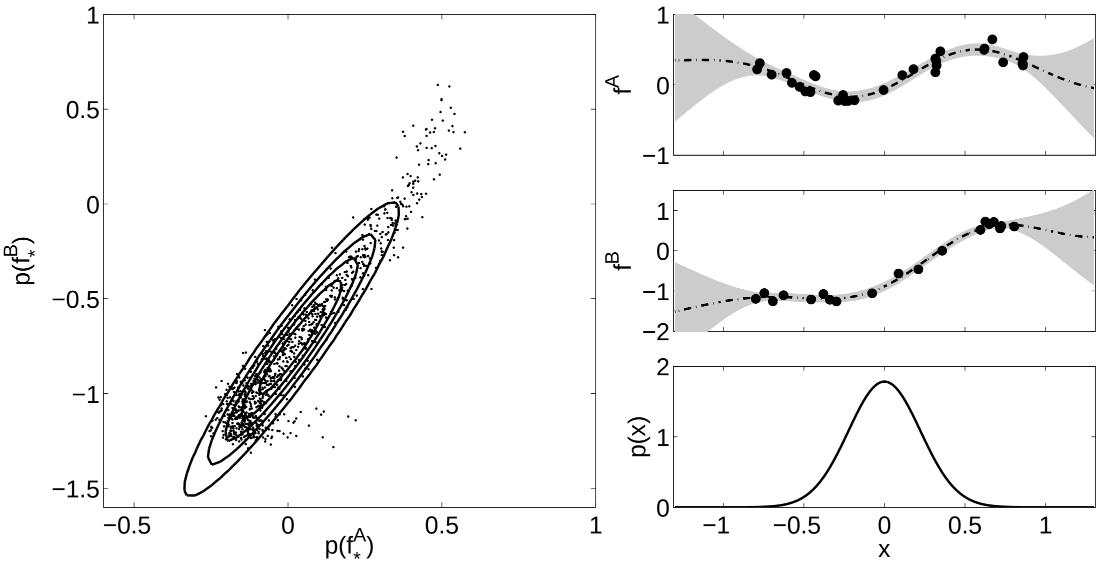


Figure 7.3.: Illustration of the covariance of predictions of two Gaussian process regression models A and B for an uncertain input x_*. On the right hand side the upper two panels show posterior Gaussian processes $f^A|\mathcal{D}^A$ and $f^B|\mathcal{D}^B$ inferred from the observations shown as points. The dashed lines describe the mean function and the grey area depicts ± 2 standard deviations respectively. The lower panel shows $\mathcal{N}(x_*|\mu_*, \sigma_*^2)$. From this distribution 1000 samples where generated and the corresponding predictive distribution of function values f_* were computed. The left hand panel shows a scatter-plot of samples from the predictive distributions, i.e. samples of $p(f_*^A, f_*^B|\mathcal{D}, \boldsymbol{\psi}, \mu_*, \sigma_*^2)$, and the bivariate normal approximation centred at $[m_1^A, m_1^B]^\top$, which shows that the covariance is well matched by the Gaussian approximation (7.30).

cess regression models with corresponding predictive distributions $p(f_*^A|\mathcal{D}^A, \boldsymbol{\theta}^A, \boldsymbol{\psi}^A, \mathbf{x}_*)$ and $p(f_*^B|\mathcal{D}^B, \boldsymbol{\theta}^B, \boldsymbol{\psi}^B, \mathbf{x}_*)$, for which we will use the shorthand notations $p(f_*^A|\mathbf{x}_*)$ and $p(f_*^B|\mathbf{x}_*)$ below.

The predictive distributions are assumed to be conditionally independent given $\mathbf{x}_*$:

$$p(f_*^A, f_*^B|\mathcal{D}^A, \boldsymbol{\theta}^A, \boldsymbol{\psi}^A, \mathcal{D}^B, \boldsymbol{\theta}^B, \boldsymbol{\psi}^B, \mathbf{x}_*) \;=\; p(f_*^A|\mathbf{x}_*)\, p(f_*^B|\mathbf{x}_*) \tag{7.29}$$

However, if we take the expectation over an uncertain input $\mathbf{x}_*$, the predictive distributions become dependent as illustrated in Figure 7.3. The covariance of predictions is

$$\begin{aligned}\operatorname{cov}(f_*^A, f_*^B) \;&=\; \int (f_*^A - m_1^A)(f_*^B - m_1^B)\, p(f_*^A|\mathbf{x}_*)\, p(f_*^B|\mathbf{x}_*)\, \mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x}_*\, df_*^A\, df_*^B \\ &=\; \boldsymbol{\beta}_A^\top \tilde{\mathbf{L}}\, \boldsymbol{\beta}_B - m_1^A m_1^B \end{aligned} \tag{7.30}$$

where m_1^A and m_1^B are the respective expectations of f_*^A and f_*^B computed by eq. (7.23). The entries of $\tilde{\mathbf{L}}$ are given element-wise by

$$\tilde{L}_{ij} = \frac{\sigma_A^2 \sigma_B^2}{\sqrt{|(\mathbf{W}_A^{-1}+\mathbf{W}_B^{-1})\boldsymbol{\Sigma}+\mathbf{I}|}} \exp\left(-\tfrac{1}{2}(\mathbf{x}_j^B-\mathbf{x}_i^A)^\top (\mathbf{W}_A+\mathbf{W}_B)^{-1} (\mathbf{x}_j^B-\mathbf{x}_i^A)\right)$$
$$\times \exp\left(-\tfrac{1}{2}(\mathbf{z}_{ij}-\boldsymbol{\mu})^\top \left(\left(\mathbf{W}_A^{-1}+\mathbf{W}_B^{-1}\right)^{-1}+\boldsymbol{\Sigma}\right)^{-1} (\mathbf{z}_{ij}-\boldsymbol{\mu})\right) \quad (7.31)$$

where $\mathbf{z}_{ij} = (\mathbf{W}_A^{-1}+\mathbf{W}_B^{-1})^{-1}(\mathbf{W}_A^{-1}\mathbf{x}_i+\mathbf{W}_B^{-1}\mathbf{x}_j)$.

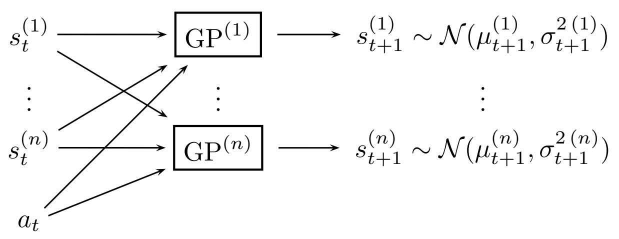
In later sections it will be of great importance that the expressions for the mean (7.23), the variance (7.28), and the covariance (7.30) can be differentiated with respect to the elements of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Note that the prediction with uncertain inputs can also be implemented for a GP model with a linear mean function. The expressions are omitted here, but can be derived analogously.

7.2.2. Model Identification & Simulation

We now turn towards the application of Gaussian process regression for *model identification* in reinforcement learning problems. Thereby we restrict ourselves to problems with continuous state spaces $\mathcal{S} \subseteq \mathbb{R}^n$, continuous actions $\mathcal{A} \subseteq \mathbb{R}$, and discrete time. Reinforcement learning problems with continuous state spaces are often similar to systems studied in control engineering. Therefore we will also use the terms control strategy and control signal synonymously to policy and action below.

Let again $\boldsymbol{s}_t \in \mathcal{S}$ denote the state of a system at time t, $a_t \in \mathcal{A}(\boldsymbol{s}_t)$ the action or control signal applied at this time, and $\boldsymbol{s}_{t+1}$ the resulting consecutive state. The time interval between two stages will be referred to as Δt. The objective of model identification is to build a model which predicts $\boldsymbol{s}_{t+1}$ given $\boldsymbol{s}_t$ and a_t based on previously observed state transitions. In this setting model identification becomes a regression problem $[\boldsymbol{s}_t, a_t] \to \boldsymbol{s}_{t+1}$ which we approach by using probabilistic GP regression models. These provide a predictive distribution $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a_t)$ which we will interpret as the agent's uncertainty about its environment.

Assuming $\mathcal{S} = \mathbb{R}^n$ we make use of n separate Gaussian process regression models, the *dynamics* GPs, each for predicting a distribution over a single variable of the consecutive state. Based on observed or generated state transitions $(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$ we find values for the hyper-parameters $\boldsymbol{\psi}$ and the noise variance σ_{n}^2 by ML-II estimation as described in Section 3.2.


$s_t^{(1)}$
$s_t^{(n)}$
a_t
$\mathrm{GP}^{(1)}$
$\mathrm{GP}^{(n)}$
$s_{t+1}^{(1)} \sim \mathcal{N}(\mu_{t+1}^{(1)}, \sigma_{t+1}^{2\,(1)})$
$s_{t+1}^{(n)} \sim \mathcal{N}(\mu_{t+1}^{(n)}, \sigma_{t+1}^{2\,(n)})$

Combining the predictive distributions of the n separate GP models the joint predictive distribution comes in the form of a multivariate normal

$$p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a_t) \;=\; \mathcal{N}(\boldsymbol{s}_{t+1}|\boldsymbol{\mu}_{t+1}, \mathbf{S}_{t+1}) \tag{7.32}$$

where $\boldsymbol{\mu}_{t+1} = [\mu_{t+1}^{(1)}, \dots, \mu_{t+1}^{(n)}]^\top$ and $\mathbf{S}_{t+1} = \operatorname{diag}(\sigma_{t+1}^{2(1)}, \dots, \sigma_{t+1}^{2(n)})$ is a diagonal covariance matrix. The use of separate regression models for each dimension is a simplification which does not exploit dependencies between state variables.

Using the dynamics GPs we can also simulate multiple steps into the future propagating the uncertainty. From an initial state $\boldsymbol{s}_t$ and given action a_t we compute a distribution over $\boldsymbol{s}_{t+1}$ (7.32). Now given an action a_{t+1} the problem is to predict the distribution of $\boldsymbol{s}_{t+2}$ if already $\boldsymbol{s}_{t+1}$ is a multivariate normal random vector. This is where the prediction for uncertain inputs can be used, which has been described in the previous section. The predictive distribution $p(\boldsymbol{s}_{t+2}|\boldsymbol{\mu}_{t+1}\mathbf{S}_{t+1}, a_{t+1})$ is approximated by a multivariate normal distribution $\mathcal{N}(\boldsymbol{s}_{t+2}|\boldsymbol{\mu}_{t+2}, \mathbf{S}_{t+2})$ by computing its first two moments analytically. The mean is computed using eq. (7.23), the diagonal elements of the covariance are given by eq. (7.28), and the off-diagonal elements are covariances of the form of eq. (7.30). Approximating the distribution by a multivariate Gaussian allows us to iterate this procedure such that we can obtain a sequence of distributions over states.

$$\begin{tikzcd}
\boldsymbol{s}_t \arrow[r] & \boxed{\text{GPs}} \arrow[r, "{\boldsymbol{\mu}_{t+1}, \mathbf{S}_{t+1}}"] & \boxed{\text{GPs}} \arrow[r, "{\boldsymbol{\mu}_{t+2}, \mathbf{S}_{t+2}}"] & \cdots \arrow[r] & \boxed{\text{GPs}} \arrow[r] & \boldsymbol{\mu}_{t+N}, \mathbf{S}_{t+N} \\
a_t \arrow[ur] & a_{t+1} \arrow[ur] & a_{t+2} \arrow[ur] & a_{t+N-1} \arrow[ur] & &
\end{tikzcd}$$

In general, let $\mathcal{N}(\boldsymbol{s}_{t+i-1}|\boldsymbol{\mu}_{t+i-1}, \mathbf{S}_{t+i-1})$ describe our uncertainty in the state of the system at time $t+i-1$ if we apply a control signal a_{t+i-1} then the distribution of $\boldsymbol{s}_{t+i}$ is approximated by $\mathcal{N}(\boldsymbol{s}_{t+i}|\boldsymbol{\mu}_{t+i}, \mathbf{S}_{t+i})$. Note that if $\boldsymbol{s}_t$ is certain, then $\mathbf{S}_{t+1}$ is diagonal and from $t+2$ onwards $\mathbf{S}_{t+i}$ may have nonzero off-diagonal entries. This technique to simulate several steps into the future will be used in Section 7.4.

7.3. Approximate Policy Iteration in Continuous Domains

Most common reinforcement learning methods aim at estimating some form of value function. In Section 7.1 we assumed finite MDPs which allowed a table based representation of the state transition probabilities, the policy, and the values. The table based representation is limited to relatively small problems and becomes impractical for large discrete domains, e.g. backgammon (Tesauro, 2002), as well as for continuous domains. In this case there are basically two different approaches: coding methods and function approximation. Unfortunately, almost all theoretical results on convergence based on exact (table) representations become invalid for these methods.

The idea of coding methods is that states which are similar with respect to their dynamical properties are grouped together, see for example Moore and Atkeson (1995).

Thereby the system is mapped to a lower dimensional or even discrete representation on which learning can be performed easier.

Function approximation techniques use a function $f(\boldsymbol{s}, \boldsymbol{\phi})$ to represent the value function $\mathsf{V}(\boldsymbol{s})$, where $\boldsymbol{\phi}$ are free parameters. Common choices for f are linear models (in some fixed set of basis functions) and neural networks. By choosing a certain class of functions we cannot hope to have the true optimal value function in this class such that an approximation error (bias) is inevitable. Temporal difference learning algorithms can be modified to be used with function approximation (Tsitsiklis and Van Roy, 1997). A temporal difference correction can be used to update the parameters $\boldsymbol{\phi}$:

$$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \alpha_t \left[r_{t+1} + \gamma f(\boldsymbol{s}_{t+1}, \boldsymbol{\phi}) - f(\boldsymbol{s}_t, \boldsymbol{\phi})\right] \nabla_{\boldsymbol{\phi}} f(\boldsymbol{s}_t, \boldsymbol{\phi}) \tag{7.33}$$

given an observed state transition $(\boldsymbol{s}_t, a_{,t}, \boldsymbol{s}_{t+1})$ and corresponding reward r_{t+1}. The update rule can be interpreted as a stochastic gradient descent on the temporal difference error. Note that the TD update rule (7.15) for table based representations can be interpreted as a special case of eq. (7.33) in which the number of parameters equals the number of states. For approximating the $\mathcal{Q}$-value the function approximator needs to be of the form $f(\boldsymbol{s}, a, \boldsymbol{\phi})$. Under certain conditions asymptotic convergence results for function-approximation based temporal difference learning have been derived, see for example Bradtke and Barto (1996) and Tsitsiklis and Van Roy (1997). However, there are simple counter examples in which function approximation leads to oscillating or even divergent behaviour (Sutton and Barto, 1998, ch. 8.5). In general, making function-approximation based reinforcement-learning work is often a complicated task including a good deal of fine-tuning and trial-and-error experiments.

7.3.1. Gaussian Process Approximate Policy Iteration

In the following we make use of a Gaussian process regression model to represent the value function V and derive an approximate *policy iteration* scheme for continuous domains. Therefore we will derive approximate *policy evaluation* and *policy improvement* methods based on Gaussian process approximations of both the value and the state transition probabilities.

Policy evaluation makes use of the Bellman equations which for continuous state spaces can be straight-forwardly generalised from the discrete version (7.6b) by substituting sums with integrals:

$$\begin{aligned} \mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}) &= \int p(\boldsymbol{s}'|\boldsymbol{s}, \mathfrak{p}(\boldsymbol{s})) \left[\mathsf{r}(\boldsymbol{s}') + \gamma \mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}')\right] \, d\boldsymbol{s}' & \text{(7.34a)} \\ &= \int \mathsf{r}(\boldsymbol{s}')\, p(\boldsymbol{s}'|\boldsymbol{s}, \mathfrak{p}(\boldsymbol{s}))\, d\boldsymbol{s}' + \gamma \int \mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}')\, p(\boldsymbol{s}'|\boldsymbol{s}, \mathfrak{p}(\boldsymbol{s}))\, d\boldsymbol{s}' \;, & \text{(7.34b)} \end{aligned}$$

where we assume for simplicity of exposition that the policy is a deterministic function $\mathfrak{p}(\boldsymbol{s}) \to a$ of the state and the expected reward r only depends on the consecutive state.

Below, GP regression models will be used for two distinct purposes: the *dynamics* GPs to approximate $p(\boldsymbol{s}'|\boldsymbol{s}, a)$ and the *value* GP for representing the value function

V. The key idea of this section is that using Gaussian process models to represent the dynamics and the value function allows us to approximate the expectations in the continuous Bellman equation (7.34b) analytically.

7.3.1.1. Approximate Policy Evaluation

We now turn to the problem of approximating the value function $\mathsf{V}(\boldsymbol{s})$ for a given policy $\mathfrak{p}$ over a continuous state space. We need access to the value function at every point in the continuous state space, but we only explicitly represent values at a finite number of *support points* $\mathsf{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m\}$ and let the Gaussian process generalise to the entire space. Let $\mathbf{v}$ be an $[m \times 1]$ vector representing the value function at the support points, such that $v_i = \mathbb{E}[\mathsf{V}(\boldsymbol{s}_i)]$ for all $\boldsymbol{s}_i \in \mathsf{S}$ and let $\boldsymbol{\sigma}^2_{\mathsf{V}} = [\sigma_1^2, \ldots, \sigma_m^2]^\top$ denote the corresponding uncertainties. The *value* GP is given by the mean and covariance function:

$$m_{\mathsf{V}}(\boldsymbol{s}) \;=\; \mathbf{k}(\boldsymbol{s})^\top \mathbf{Q}_{\mathsf{V}}^{-1}\mathbf{v} \tag{7.35a}$$

$$k_{\mathsf{V}}(\boldsymbol{s}, \tilde{\boldsymbol{s}}) \;=\; k(\boldsymbol{s}, \tilde{\boldsymbol{s}}) - \mathbf{k}(\boldsymbol{s})^\top \mathbf{Q}_{\mathsf{V}}^{-1}\mathbf{k}(\tilde{\boldsymbol{s}}) \tag{7.35b}$$

where $\mathbf{Q}_{\mathsf{V}} = \mathbf{K} + \mathrm{diag}(\boldsymbol{\sigma}^2_{\mathsf{V}})$, $\mathbf{K}$ denotes the $[m \times m]$ prior covariance matrix computed from the support points, and $\boldsymbol{s}, \tilde{\boldsymbol{s}} \in \mathcal{S}$. In order to evaluate (7.34b) analytically, the prior covariance function $k(\boldsymbol{s}, \tilde{\boldsymbol{s}})$ has to be a squared exponential as given by eq. (7.21).

In *policy evaluation*, as described in Section 7.1.3, the Bellman equation is used as an update rule. Analogously, we now use the continuous Bellman equation for updating the value $\mathbf{v}$ at the support points.

Evaluating the continuous Bellman equation (7.34) for a given combination of state $\boldsymbol{s}$ and action a involves two steps. First the distribution $p(\boldsymbol{s}'|\boldsymbol{s}, \mathfrak{p}(\boldsymbol{s})) \approx \mathcal{N}(\boldsymbol{s}'|\boldsymbol{\mu}, \mathbf{S})$ is computed using the dynamics GPs. Secondly the value is estimated by computing the two expectations in eq. (7.34b). The first integral gives the expected reward

$$r_i \;=\; \int \mathsf{r}(\boldsymbol{s}')\,\mathcal{N}(\boldsymbol{s}'|\boldsymbol{\mu}, \mathbf{S})\,d\boldsymbol{s}' \tag{7.36}$$

where the expectation is taken with respect to the uncertainty in the consecutive state. For simple (e.g. polynomial or Gaussian) reward functions the expectation can be computed directly, for instance using the integrals (B.26) or (B.24). For more complex reward functions we may approximate it using, e.g., a Taylor expansion as will be exemplified in Section 7.4.2. For simplicity we have assumed that the reward function is deterministic and known, but it is also possible to use a Gaussian process regression model for the rewards in which case the integral (7.36) can be solved using eq. (7.23).

The second integral of eq. (7.34b) involves an expectation over the value GP, which can be done in closed form as described in Section 7.2.1. So we obtain the update for the value at support point $\boldsymbol{s}_i$:

$$v_i \;\leftarrow\; r_i + \gamma \int \mathsf{V}(\boldsymbol{s}')\,\mathcal{N}(\boldsymbol{s}'|\boldsymbol{\mu}, \mathbf{S})\,d\boldsymbol{s}' \;=\; r_i + \gamma \mathbf{l}^\top \mathbf{Q}_{\mathsf{V}}^{-1}\mathbf{v} \tag{7.37}$$

where $\mathbf{l}$ is implicitly defined by eq. (7.23). Equation (7.37) could be used for iterative policy evaluation, updating one value at a time. However, note that eq. (7.37) gives rise to a set of $|\mathcal{S}|$ *linear* simultaneous equations in $\mathbf{v}$, which we can solve explicitly:

$$\mathbf{v} \;=\; \mathbf{r} + \gamma \mathbf{W}\mathbf{Q}_{\mathsf{V}}^{-1}\mathbf{v} \quad \Longrightarrow \quad \mathbf{v} \;=\; \left(\mathbf{I} - \gamma \mathbf{W}\mathbf{Q}_{\mathsf{V}}^{-1}\right)^{-1}\mathbf{r} \tag{7.38}$$

where $\mathbf{W}$ is a matrix such that its ith row is equal to the $\mathbf{l}$ vector corresponding to the ith support point as in eq. (7.37). Note the similarity of eq. (7.38) to the table based eq. (7.12).

It should be stressed that—unfortunately—we are unable to perform proper Bayesian inference about the value function, because we cannot state a corresponding likelihood. However, the uncertainty about $\mathbf{v}$ can be modelled by $\boldsymbol{\sigma}_{\mathsf{V}}^2$. One approach is to interpret the rôle of the value GP as a noise free interpolation of the values $\mathbf{v}$, as illustrated in Figure 3.1 on page 18, and therefore to set its noise parameter $\boldsymbol{\sigma}_{\mathsf{V}}^2$ to a fixed small positive value (to avoid numerical problems). Otherwise one can compute the variance of v_i by approximating the variance of (7.37), which can be computed by eq. (7.28). This will be further discussed at the end of this chapter.

7.3.1.2. Policy Improvement

Above we described how to estimate the value function for a given policy $\mathfrak{p}$, i.e. approximate policy evaluation using Gaussian process models. Now given a GP representation of the value function and the dynamics GPs we can implement policy improvement by acting greedily, thereby defining an implicit policy:

$$\mathfrak{p}^{+}(\boldsymbol{s}) \leftarrow \underset{a \in \mathcal{A}(\boldsymbol{s})}{\operatorname{argmax}} \int p(\boldsymbol{s}'|\boldsymbol{s}, a) \left[\mathsf{r}(\boldsymbol{s}') + \gamma \mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}')\right] \, d\boldsymbol{s}' \,. \tag{7.39}$$

In a policy improvement step we have to compute the greedy action for all support states which gives rise to $|\mathcal{S}|$ one-dimensional optimisation problems (when the possible actions a are scalar). As we were able to compute the right hand side integral analytically, see eq. (7.37), we can also compute the gradient of this quantity w.r.t. the action. Hence, we can implement the optimisation problem eq. (7.39) using gradient based methods. In the examples below we will see that application-specific constraints can often be reformulated as constraints in the above optimisation problem, for example constraints on a being in certain limits.

7.3.1.3. Approximate Gaussian Process Policy Iteration

We now combine *policy evaluation* and *policy improvement* into *policy iteration* in which both steps alternate until a stable configuration is reached, see Algorithm 5. Thus given observed state transitions and a reward function (or observations of rewards) we can compute an estimate of the continuous value function and thereby an implicitly defined greedy policy.

A practical problem of the approach is that function approximation errors can occur, which then can propagate over the whole function and in the worst case lead to divergent behaviour. Since the value function is updated based on evaluations of itself, errors can be distributed, which is a common problem of function approximation methods in reinforcement learning, see also eq. (7.33). If the value of a state $\boldsymbol{s}'$ is overestimated, this will lead to an overestimation of the values of all states $\boldsymbol{s}$ that project to it $\boldsymbol{s} \to \boldsymbol{s}'$ given the current policy. Unfortunately these kinds of errors are difficult to detect. In practice it is therefore advisable to monitor the maximum value of $\mathbf{v}$ in every iteration. An upper bound for the maximum value of $\mathbf{v}$ is given by the discounted sum of the maximum reward given the number of iterations. Note that the effect is down-weighted by the discount rate. It has been observed empirically that smaller values of the discount rate γ help to limit the effects of this problem. Furthermore, the policy improvement step involves non-convex optimisation problems (7.39). If only a sub-optimal action is found, this can also lead to an erroneous update.

Algorithm 5 Gaussian process approximate policy iteration, batch version

```
1. Given: Observations of system dynamics of the form $(\boldsymbol{s}, a, \boldsymbol{s}')$ for a fixed time
interval $\Delta t$, discount factor $\gamma$ and reward function $\mathsf{r}$
2. Model Identification: ML-II estimation of the parameters of the dynamics GPs
in order to obtain $p(\boldsymbol{s}'|\boldsymbol{s}, a) \approx \mathcal{N}(\boldsymbol{s}'|\boldsymbol{\mu}, \mathbf{S})$.
3. Initialise Value Function: Choose a set $\mathsf{S} = \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_m\}$ of $m$ support points
and initialise $v_i \leftarrow \mathsf{r}(\boldsymbol{s}_i)$. Estimate Gaussian process hyper-parameters $\boldsymbol{\psi}$ for repre-
senting $\mathsf{V}(\boldsymbol{s})$ using ML-II.
4. Policy Iteration:
repeat
  for all $\boldsymbol{s}_i \in \mathsf{S}$ do
    Find action $a_i$ according to implicit policy by solving equation (7.39) subject to
    problem specific constraints.
    Compute $\mathcal{N}(\boldsymbol{s}_i'|\boldsymbol{s}_i, a_i)$ using the dynamics Gaussian processes.
    Solve equation (7.36) in order to obtain $r_i$.
    Compute $i$th row of $\mathbf{W}$ as in equation (7.37).
  end for
  $\mathbf{v} \leftarrow (\mathbf{I} - \gamma \mathbf{W} \mathbf{Q}_{\mathsf{V}}^{-1})^{-1} \mathbf{r}$
  Update Gaussian process hyper-parameters for representing $\mathsf{V}(\boldsymbol{s})$ to fit the new $\mathbf{v}$
  by ML-II.
until stabilisation of $\mathbf{v}$
```

The selection of the support points remains to be determined. In principle, the support states should be placed such that the true value function—and the intermediate ones during policy iteration—can be approximated by the value GP as well as possible. However, it is not obvious how to find these optimal locations *a priori*. In the experiments described in the following sections, we simply use a regular grid of support points in the relevant region of the state space.

7.3.2. Estimating the $\mathcal{Q}$-Value Function

In the policy iteration scheme we exploited that if both the dynamics of the environment and the state-value function V are represented by Gaussian process models, we can compute the expected value of the consecutive state. A variation of this setting is to estimate the state-action value function $\mathcal{Q} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as defined by eq. (7.7). The continuous version of the $\mathcal{Q}$-function comes in the form

$$\mathcal{Q}^{\mathfrak{p}}(\boldsymbol{s}, a) \;=\; \int p(\boldsymbol{s}'|\boldsymbol{s}, a) \; \left[\mathsf{r}(\boldsymbol{s}') + \gamma \mathsf{V}^{\mathfrak{p}}(\boldsymbol{s}')\right] \, d\boldsymbol{s}' \tag{7.40}$$

where again we made the assumption that the reward depends on the consecutive state only. Using the greedy policy $\mathfrak{p}^+$ and the relation (7.8) between the V and $\mathcal{Q}$ function we obtain:

$$\mathcal{Q}(\boldsymbol{s}, a) \;=\; \int p(\boldsymbol{s}'|\boldsymbol{s}, a) \left[\mathsf{r}(\boldsymbol{s}') + \gamma \max_{a' \in \mathcal{A}(\boldsymbol{s}')} \mathcal{Q}(\boldsymbol{s}', a')\right] d\boldsymbol{s}' \tag{7.41}$$

which defines a self-consistency equation for the optimal $\mathcal{Q}$-function.

In the following we use a Gaussian process regression model to represent $\mathcal{Q}(\boldsymbol{s}, a)$ and eq. (7.41) will be used iteratively as an update rule for estimating the optimal $\mathcal{Q}$-values at the support points. In this setting the support points come in the form of state-action pairs $\mathsf{S} = \{(\boldsymbol{s}, a)_i | i = 1, \ldots, m\}$ and let $\mathbf{q}$ denote the $[m \times 1]$ vector of corresponding $\mathcal{Q}$-values. The algorithm iteratively updates the estimated $\mathcal{Q}$-values at the support states

$$q_i \leftarrow r_i + \gamma \max_{a' \in \mathcal{A}(\boldsymbol{s}')} \int \mathcal{Q}(\boldsymbol{s}', a') \, \mathcal{N}(\boldsymbol{s}' | \boldsymbol{\mu}_i, \mathbf{S}_i) \, d\boldsymbol{s}' \tag{7.42}$$

where r_i is the expected immediate reward for the ith state-action pair in the support set.

For each support point $(\boldsymbol{s}, a)_i$ we first compute the predictive distribution over the consecutive state $p(\boldsymbol{s}' | \boldsymbol{s}_i, a_i) = \mathcal{N}(\boldsymbol{s}' | \boldsymbol{\mu}_i, \mathbf{S}_i)$ using the dynamics GPs. Note that these predictive distributions have to be computed only once for each support point. Then the expectation over the reward function can be computed

$$r_i \;=\; \int \mathsf{r}(\boldsymbol{s}') \, \mathcal{N}(\boldsymbol{s}' | \boldsymbol{\mu}_i, \mathbf{S}_i) \, d\boldsymbol{s}' \;, \tag{7.43}$$

which also do not change during the learning process. We then have to solve for the value of the consecutive state under the greedy policy. For each support point we have to find the value of

$$\max_{a' \in \mathcal{A}(\boldsymbol{s}')} \int \mathcal{Q}(\boldsymbol{s}', a') \, \mathcal{N}(\boldsymbol{s}' | \boldsymbol{\mu}_i, \mathbf{S}_i) \, d\boldsymbol{s}' \;, \tag{7.44}$$

which is an expectation of a Gaussian process prediction for an uncertain input as described in Section 7.2.1. Technically, we solve the integral

$$\int \mathcal{Q}(\boldsymbol{s}', a') \, \mathcal{N}\!\left(\begin{bmatrix} \boldsymbol{s}' \\ a' \end{bmatrix} \middle| \begin{bmatrix} \boldsymbol{\mu}_i \\ a' \end{bmatrix}, \begin{bmatrix} \mathbf{S}_i & 0 \\ 0 & 0 \end{bmatrix} \right) d\boldsymbol{s}' \, da' \tag{7.45}$$

by using eq. (7.23) and compute the gradient with respect to a'. Note that the singular covariance matrix in eq. (7.45) has not to be inverted directly in eq. (7.23), so we get away with this trick. Thereby we can use a conjugate gradient optimisation routine to solve the individual problems (7.44). The procedure of approximating the optimal $\mathcal{Q}$-function by a Gaussian process is summarised in Algorithm 6.

Algorithm 6 Gaussian process approximate $\mathcal{Q}$-Learning, batch version

```
1. Given: Observations of system dynamics of the form $(\boldsymbol{s}, a, \boldsymbol{s}')$ for a fixed time
interval $\Delta t$, discount factor $\gamma$ and reward function $\mathsf{r}$.
2. Model Identification: ML-II estimation of the parameters of the dynamics GPs
in order to obtain $p(\boldsymbol{s}'|\boldsymbol{s}, a) \approx \mathcal{N}(\boldsymbol{s}'|\boldsymbol{\mu}, \mathbf{S})$.
3. Initialise Q-value GP: Choose a set of support points $\mathsf{S} = \{(\boldsymbol{s}, a)_i | i = 1, \dots, m\}$,
compute predictive distributions $\mathcal{N}(\boldsymbol{s}'_i|\boldsymbol{\mu}_i, \mathbf{S}_i)$ for all support points using the dynamics
GPs and store parameters. Compute $\mathbf{r}$ according to (7.43) and initialise $\mathbf{q} \leftarrow \mathbf{r}$.
4. Estimation of Q-function:
repeat
  for all $(\boldsymbol{s}, a)_i \in \mathsf{S}$ do
      $q_i \leftarrow r_i + \gamma \max_{a' \in \mathcal{A}(\boldsymbol{s}')} \int \mathcal{Q}(\boldsymbol{s}', a') \mathcal{N}(\boldsymbol{s}'|\boldsymbol{\mu}_i, \mathbf{S}_i)\, d\boldsymbol{s}'$
  end for
  Update GP hyper-parameters for representing $\mathcal{Q}(\boldsymbol{s}, a)$ to fit the new $\mathbf{q}$
until stabilisation of $\mathbf{q}$
```

7.3.3. The Mountain Car Problem

For the first experiments the well-known mountain car problem, as described by Moore and Atkeson (1995), will be used where the state-space $\mathcal{S}$ is only two-dimensional. The setting consists of a friction-less, point-like, unit mass car in a valley. The state of the system $\boldsymbol{s} = [x, \dot{x}]^\top$ is described by the position of the car and its speed which are constrained to $-1 \leq x \leq 1$ and $-2 \leq \dot{x} \leq 2$ respectively.

For control a horizontal force F in the range $-4 \leq F \leq 4$ can be applied in order to drive the car up into the target region which in the formulation of Moore and Atkeson (1995) is a rectangle in state space such that $0.5 \leq x \leq 0.7$ and $-0.1 \leq \dot{x} \leq 0.1$. Note that the admissible range of forces is not sufficient to drive up the car greedily from the initial state $\boldsymbol{s}_0 = [-0.5, 0]^\top$. Therefore, a strategy has to be found which utilises the landscape in order to accelerate up the slope, which gives the problem its non-minimum phase character, i.e. the problem cannot be solved by greedily maximising the immediate reward.

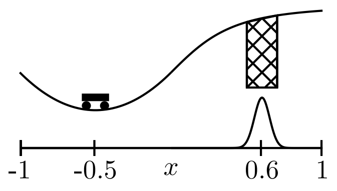


Figure 7.4.: Illustration of the mountain car problem.

For more details about the dynamics of the system see Appendix C.1.

At first we demonstrate how the approximate policy iteration scheme can be used to estimate the optimal state-value function V. For model identification 50 state transitions $(\boldsymbol{s}, a, \boldsymbol{s}')$ are simulated for $\Delta t = 0.2$sec. After ML-II estimation of parameters the *dynamics* GPs approximate the dynamics of the system within root mean squared errors of 0.02 for predicting x and 0.2 for predicting $\dot{x}$.

Having a model of the system dynamics, the other necessary element we need to specify is a reward function. In the formulation by Moore and Atkeson (1995) the reward is equal to 1 if the car is in the target region and 0 elsewhere. For convenience we approximate this cube by a Gaussian proportional to $\mathcal{N}([0.6, 0]^\top, 0.05^2\mathbf{I})$ with maximum reward 1 as sketched in Figure 7.4. We now can solve the update equation (7.37) and also evaluate its gradient with respect to F. This enables us to efficiently solve the optimisation problem eq. (7.39) subject to the constraint $-4 \leq F \leq 4$. States outside the feasible region $-1 \leq x \leq 1$ and $-2 \leq \dot{x} \leq 2$ are assigned zero value and reward.

As support points S for the state-value function V we simply put a regular $[19 \times 19]$ grid onto the state-space and initialise the value function with the immediate rewards for these states, see Figure 7.5(a). The standard deviation of the noise of the *value* GP representing $\mathsf{V}(\boldsymbol{s})$ is set to $\sigma_\mathsf{V} = 0.01$, and the discount factor to $\gamma = 0.9$. Following the policy iteration scheme as given by Algorithm 5, we estimate the value of all support points following the implicit policy (7.39) w.r.t. the initial value function, see Figure 7.5(a). We then evaluate this policy and obtain an updated value GP shown in Figure 7.5(b) where all points which can expect to reach the reward region in one time step gain value. If we iterate this procedure two times, we obtain a value function as shown in Figure 7.5(c) in which the state space is already well organised. After six policy iteration steps the value function and therefore the implicit policy becomes stable, see Figure 7.5(d).

In Figure 7.6(a) a GP based state-transition diagram is shown, in which each support point $\boldsymbol{s}_i$ is connected to its predicted (mean) consecutive state $\boldsymbol{s}'_i$ when following the approximate optimal policy. For some of the support points the model correctly predicts that the car will leave the feasible region, no matter what force $|F| \leq 4$ is applied, which corresponds to areas with zero value in Figure 7.5(d).

If we control the car from $\boldsymbol{s}_0 = [-0.5, 0]^\top$ according to the greedy policy, the car gathers momentum by first accelerating left before driving up into the target region where it is balanced as illustrated in Figure 7.6(b). It shows that the 50 random examples of the system dynamics are sufficient for this task. The control policy found is probably very close to the optimally achievable one. Note that in the experiment shown in Rasmussen and Kuss (2004) the time interval was $\Delta t = 0.3$ and $\gamma = 0.8$ was used which gives similar results.

We also estimated the $\mathcal{Q}$-value function as described in Section 7.3.2 for the mountain car system. The support points S were placed on a regular $[13 \times 13 \times 10]$ grid of $x \times \dot{x} \times F$ values. The discount rate γ was set to 0.9 and we used the same dynamics GPs as in the previous experiment. After approximately 20 iterations the $\mathbf{q}$ values stabilised and the estimated $\mathcal{Q}$-function is illustrated in Figure 7.7. The value function for the greedy policy with respect to the $\mathcal{Q}$-value GP is similar to the value function found by

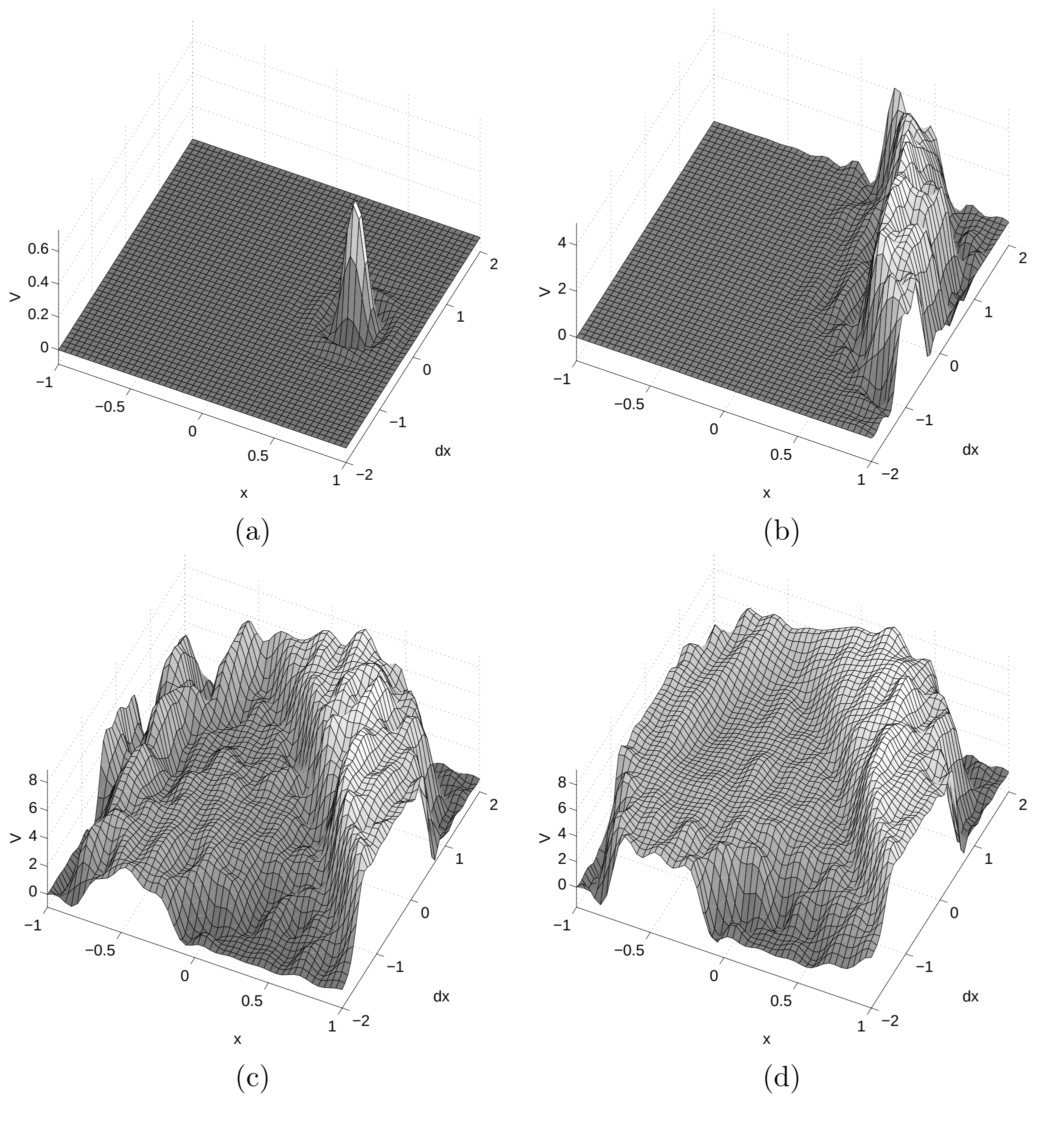


Figure 7.5.: Illustration of GP approximation to the value function V for the mountain car problem using the policy iteration scheme. Shown are the mean of the value GP for the mountain car example after initialisation by the reward (a), after the first iteration over $\mathcal{S}$ (b), and a nearly stabilised value function after 3 iterations (c). Panel (d) shows the final value function after 6 policy improvements over $\mathcal{S}$ where V has stabilised. Compare to Sutton and Barto (1998, Figure 8.10).

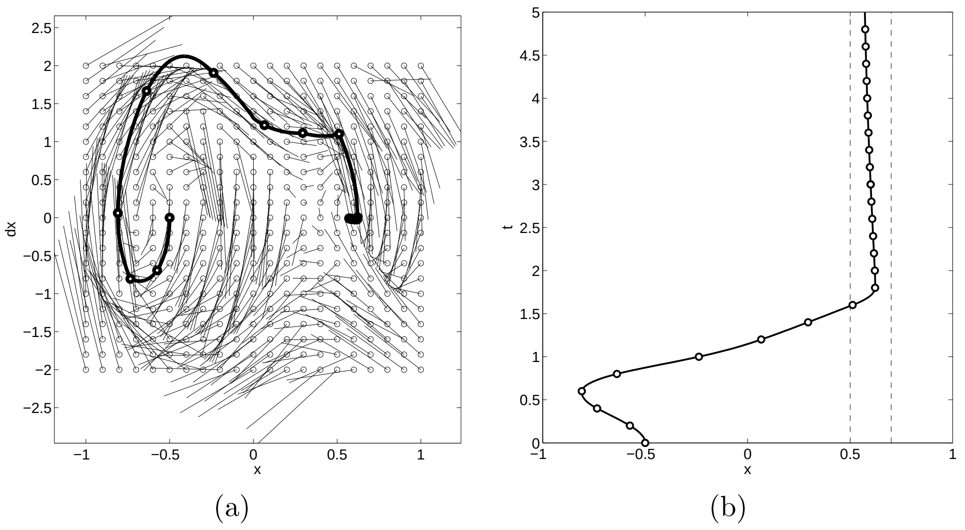


Figure 7.6.: Illustration of the greedy policy with respect to the value function shown in Figure 7.5(d). Panel (a) shows a state transition diagram illustrating the implicit policy on a grid of states. The black lines connects states $\boldsymbol{s}$ and the respective $\mathbb{E}[\boldsymbol{s}']$ estimated by the dynamics GPs when following the greedy policy. The bold trajectory describes the state of the system when controlled according to the implicit policy (7.39). Note that the temporary violation of the constraint $\dot{x} < 2$ remains unnoticed using time intervals of $\Delta t = 0.2$. Panel (b) shows the position x of the car when controlled according to the greedy policy where the circles mark the $\Delta t = 0.2$ second time steps.

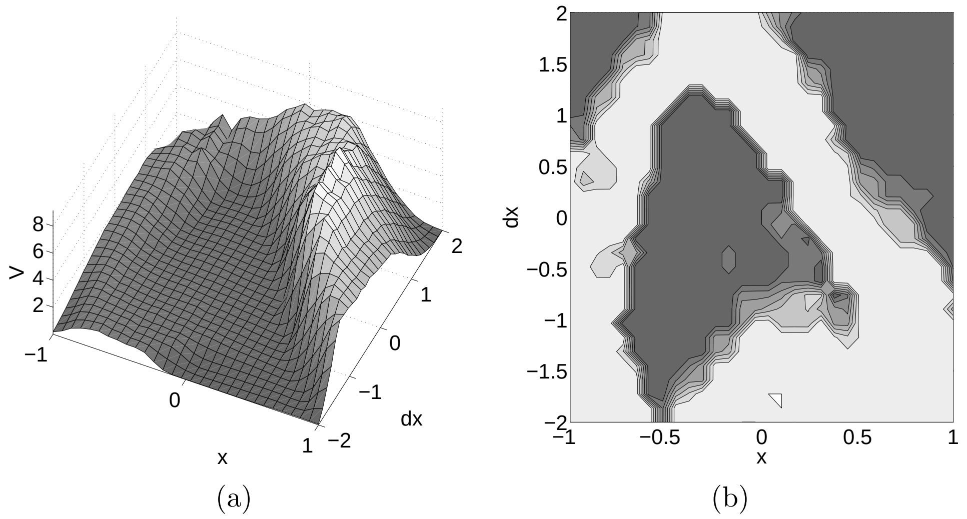


Figure 7.7.: Gaussian process approximate $\mathcal{Q}$-Learning for the mountain car problem. Panel (a) shows the state-value function of the greedy policy $\mathsf{V}(\boldsymbol{s}) = \max_a \mathcal{Q}(\boldsymbol{s}, a)$ after 20 iterations of Algorithm 6. Panel (b) illustrates the greedy policy $\mathfrak{p}(\boldsymbol{s}) = \operatorname{argmax}_a \mathcal{Q}(\boldsymbol{s}, a)$ where the gray-scale encodes the strength and direction of the control signal F from $4N$ (white) to $-4N$ (dark).

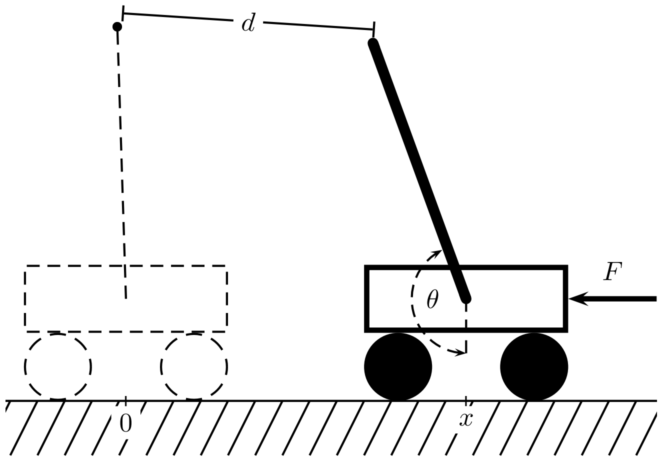


Figure 7.8.: Illustration of the inverted pendulum problem. The state of the system at time t is described by its horizontal position x, its velocity $\dot{x}$, the angle of the pendulum θ and its angular velocity $\dot{\theta}$. The aim is to stand at $x = 0$ with the pendulum upright $\theta = \pi$.

approximate policy iteration. Using the greedy policy (7.17) shown in Figure 7.7(b) the car can be driven up the hill and balanced in the target area.

7.3.4. The Inverted Pendulum

Another frequently studied system in reinforcement learning is the *inverted pendulum* which consists of a pendulum attached to a cart which can only be accelerated horizontally. The state of the system is a quadruple $\boldsymbol{s} = [x, \dot{x}, \theta, \dot{\theta}]^\top$ of the cart's position x, its velocity $\dot{x}$, the angle of the pendulum θ relative to pointing exactly downwards, and the angular velocity $\dot{\theta}$, see Figure 7.8. The control signal is a horizontal force F which can be chosen every $\Delta t = 0.2$sec for which the force is applied constantly. Learning to balance the pendulum in an upright position is a nontrivial reinforcement learning problem, if the agent does not know the dynamics. An even more challenging task is to start with the pendulum hanging downwards, such that it has to be swung up first.

For model identification we sampled $m = 300$ states uniformly from a constrained region of the state space where the cart's position is in the range $-1 \leq x \leq 1$, its velocity is $-3 \leq \dot{x} \leq 3$, the angle of the pendulum is $0 \leq \theta \leq 2\pi$, and the angular velocity is $-8 \leq \dot{\theta} \leq 8$. For each of these states we drew a force from a uniform distributions on $-4 \leq F \leq 4$ and used a simplified model of the system dynamics (see Appendix C.1 for the ODE) to simulate the system $\Delta t = 0.2$sec forward in time to obtain samples of the form $(\boldsymbol{s}, a, \boldsymbol{s}')_i$ for $i = 1, \dots, 300$.

Using the inverted pendulum system we experimented with two tasks of different difficulty. The first task is to swing up the pendulum and balance it at $x = 0$. Accordingly the reward function was a Gaussian centred on $\theta = \pi$ and $x = 0$ scaled such that the maximum reward is 1. We tried to make the approximate Gaussian process policy iteration algorithm work as described by Algorithm 5. Unfortunately we were unable to estimate a value function such that the corresponding policy could be used to swing up and balance the pendulum at the desired position $x = 0$. We tried to understand the reasons for this shortcoming by conducting several experiments.

For the inverted pendulum the state space is four dimensional and the optimal value function is a complex object which is non-stationary, i.e. it has different local properties in different parts of the state space, in particular it has discontinuities. Only in a tiny sub-region of the state space the pendulum can be balanced. At the edges of this region the value function will be discontinuous because a whole "swing" would be necessary before the pole can be balanced again. Outside the balancing region forces must be applied that push the cart towards its target position and swing it up. Once the pendulum reaches a state in the balancing region it must then be controlled very accurately. While the dynamics of the system can be approximated well, representing this non-stationary value function is difficult and a Gaussian process regression model with a covariance of the form (7.21) seems to be unsuited.

It might be that the optimal value function of the inverted pendulum problem could be approximated sufficiently well if the set of support points was large enough and the points were placed well. However, even using a regular grid of 10 values per dimension we did not obtain a value function that could swing up the pendulum, although balancing was possible if the support points were placed in the balancing region. As will be shown in Section 7.4.2 swinging up the pendulum and balancing it at $x = 0$ can be implemented, but using a different approach.

We then simplified the task by ignoring the position x of the cart. The task becomes to swing up the pole and balance it independent of the position and the speed of the cart. Accordingly the reward function was chosen to be a Gaussian centred on $\theta = \pi$ but which is independent of all other components of the state. Furthermore, larger forces $-8 \leq F \leq 8$ were allowed such that the task can be solved in less time steps. The discount factor was set to $\gamma = 0.9$ and as support points a regular $[5 \times 9 \times 9]$ grid in the now three-dimensional state space was used, where the speed was $-3 \leq \dot{x} \leq 3$, the angle $0 \leq \theta < 2\pi$, and the angular speed $-8 \leq \dot{\theta} \leq 8$.

We then used Algorithm 5 to approximate the optimal state-value function. In multiple simulations we obtained some solutions for which the greedy policy was able to swing up the pendulum, e.g. the one shown in Figure 7.9, but we also observed estimates that failed.

7.4. Finite Horizon Planning under Uncertainty

In every stage of the decision process the agent faces the problem of choosing an action given its current state of information. The approaches described above aimed at solving this problem by estimating value functions in Markov decision processes of infinite length for an infinite planning horizon. But estimating an optimal value function involves solving this problem for all states, not just for the states that are actually relevant for the agent.

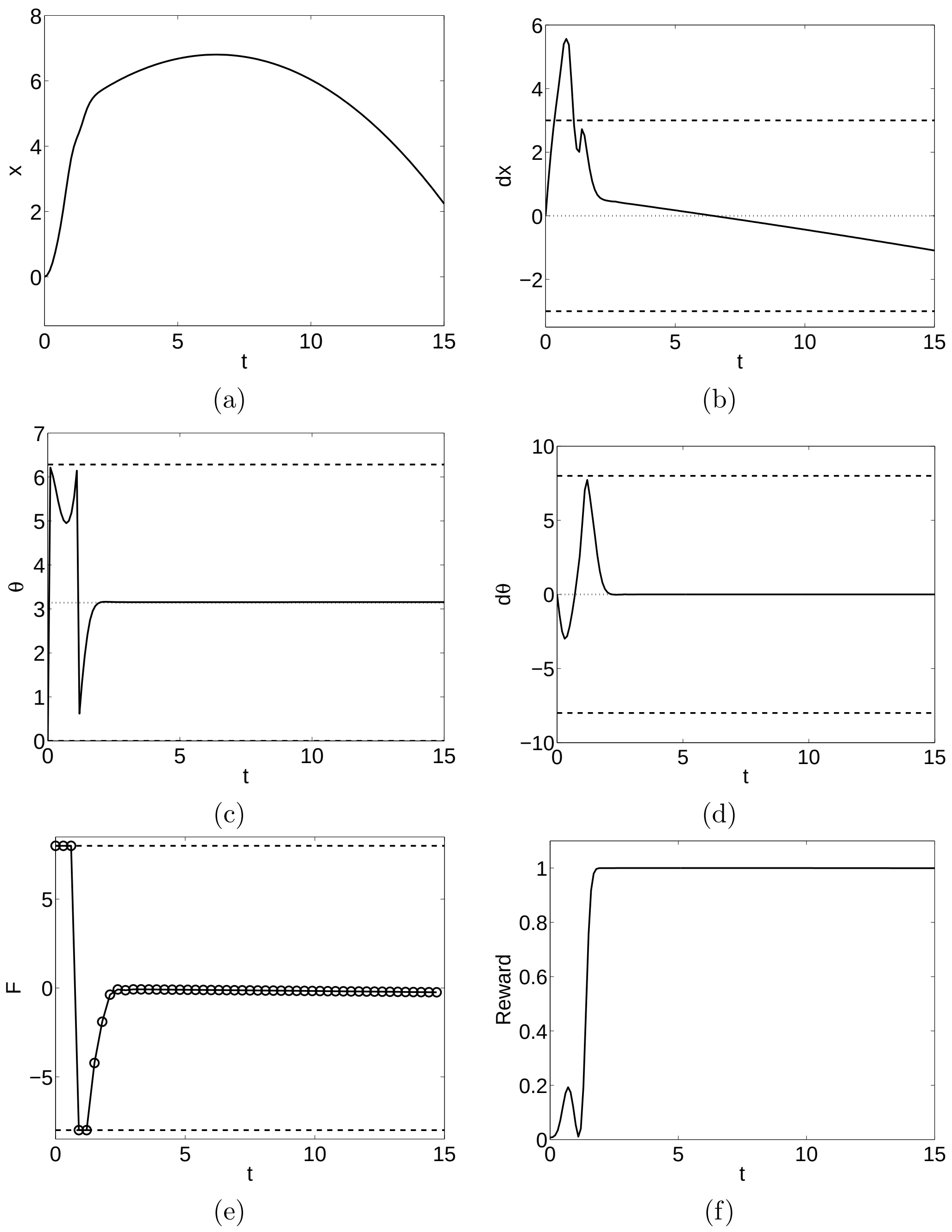


Figure 7.9.: Controlling the simplified inverted pendulum system following the greedy policy. Panels (a–d) display the state variables $\boldsymbol{s}_t = [x_t, \dot{x}_t, \theta_t, \dot{\theta}_t]^\top$ over time when applying control signals according to the greedy policy. The dashed lines mark the regions in which the training samples were generated from. Panel (e) shows the applied forces which were constrained to $-8 \leq F_t \leq 8$ (dashed lines) and Panel (f) shows the corresponding Rewards.

In this section we follow a different approach which is motivated by direct application of Bayesian decision theory as described in Section 2.2. The idea is that given a model of the environment, the agent can simulate the effects of actions several steps into the future. Given a (simple) loss function the agent can estimate the effects of a sequence of actions and choose the action that minimises the expected risk. Note that we do not aim at solving the exploration vs. exploitation problem but only to exploit the current understanding of the environment.

According to Bayesian decision theory an action is chosen which minimises the expected loss, i.e. the risk. We therefore use a loss function instead of a reward function below, but by setting the loss to be the negative reward the two views are equivalent. Let $\mathsf{L}(\boldsymbol{s}_t, \mathbf{t}_t)$ denote the loss associated with a discrepancy of the state $\boldsymbol{s}_t$ and a target state $\mathbf{t}_t \in \mathcal{S}$.

Given a probabilistic model of the state transition probabilities, a greedy approach would be to minimise the expected loss (risk) in each time step:

$$a_t \;=\; \underset{a\in\mathcal{A}(\boldsymbol{s}_t)}{\operatorname{argmin}}\, \mathsf{R}(\mathbf{t}_{t+1}, \boldsymbol{s}_t, a) \;=\; \underset{a\in\mathcal{A}(\boldsymbol{s}_t)}{\operatorname{argmin}} \int \mathsf{L}(\boldsymbol{s}_{t+1}, \mathbf{t}_{t+1})\, p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a)\, d\boldsymbol{s}_{t+i} \tag{7.46}$$

averaged over the uncertainty $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a)$ in the consecutive state. However, the kind of problems studied in reinforcement learning can typically not be solved without accepting an increase in losses before getting the system close to the target state. Therefore, a one-step greedy approach (7.46) will fail and instead one has to consider a longer planing horizon. Again we will follow the idea of minimising the expected loss, but now we use the model to simulate several steps forward in time and optimise over a sequence of future actions.

Consider the sum of discounted expected losses accumulated over N consecutive time steps given a sequence of actions $a_t, \ldots, a_{t+N-1}$, an initial state $\boldsymbol{s}_t$, and target states $\mathbf{t}_{t+1}, \ldots, \mathbf{t}_{t+N}$. The greedy approach (7.46) can be seen as a special case where $N = 1$. Let $\mathfrak{R}$ denote the discounted sum of expected losses over N steps

$$\begin{aligned}\mathfrak{R}(a_t, \ldots, a_{t+N-1}, \boldsymbol{s}_t) &= \sum_{i=1}^{N} \gamma^i \int_{\mathcal{S}} \mathsf{L}(\boldsymbol{s}_{t+i}, \mathbf{t}_{t+i})\, p(\boldsymbol{s}_{t+i}|a_t, \ldots, a_{t+i-1}, \boldsymbol{s}_t)\, d\boldsymbol{s}_{t+i} \\ &= \sum_{i=1}^{N} \gamma^i \mathsf{R}_{t+i} \end{aligned} \tag{7.47}$$

where $\gamma > 0$ is a discount factor and R is the risk, i.e. the expected loss.

In Section 7.2.2 it has been described how a Gaussian process model of the agent's environment can be used to simulate the effect of a sequence of actions. Thereby the distribution $p(\boldsymbol{s}_{t+i}|a_t, \ldots, a_{t+i-1}, \boldsymbol{s}_t)$ can be approximated by $\mathcal{N}(\boldsymbol{s}_{t+i}|\boldsymbol{\mu}_{t+i}, \mathbf{S}_{t+i})$ for $i = 1, \ldots, N$. Note that the discounted sum of expected losses (7.47) is equivalent to the utility function (7.2), by setting the reward to be the negative loss.

In order to find the action in a given state, we now minimise $\mathfrak{R}$ over the sequence of N

future actions. So at each time step t we face an N dimensional optimisation problem

$$a_t \; = \; \operatorname*{argmin}_{a_t} \; \min_{a_{t+1},\dots,a_{t+N-1}} \; \mathfrak{R}(a_t, \dots a_{t+N-1}, \boldsymbol{s}_t) \tag{7.48}$$

subject to problem specific constraints. Having found a sequence of actions that minimises $\mathfrak{R}$ we apply the first action a_t, observe $\boldsymbol{s}_{t+1}$, and repeat the procedure, which is known as a *receding horizon strategy*, see Algorithm 7.

Algorithm 7 Receding Finite Horizon Planning

```
**Given:** Initial state $\boldsymbol{s}_0$, target trajectory $\mathbf{t}_t$, discount factor $\gamma$, horizon $N$, examples
of system dynamics $(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})_i$ $i = 1, \dots, m$ for fixed $\Delta t$, loss function $\mathsf{L}(\boldsymbol{s}_t, \mathbf{t}_t)$
**1. Model Identification** Find ML-II parameters for dynamics GPs.
**2. Control**
Set $t \leftarrow 0$
**loop**
    Find $a_t$ by solving $a_t = \operatorname*{argmin}_{a_t} \min_{a_{t+1},\dots,a_{t+N-1}} \mathfrak{R}(a_t, \dots, a_{t+N-1}, \boldsymbol{s}_t)$
    Apply $a_t$ for $\Delta t$ and observe $\boldsymbol{s}_{t+1}$
    $t \leftarrow t + 1$
**end loop**
```

The discount factor can be a used to influence the strategy and has to be chosen with respect to the planing horizon N. Small values $\gamma \ll 1$ favour strategies which bring the system into states associated with low losses quickly but may neglect future consequences thereof. Too small values of γ may also bar the agent from accepting an increase in losses before reaching states associated with relatively low losses. The choice of γ must also be related to the uncertainty of the model's prediction as it grows with N. When predicting into the future, the uncertainty typically grows with the number of steps and likewise the expected losses. In order to emphasise that it is important to achieve low losses in later stages a value $\gamma > 1$ can be used to anticipate the increase in uncertainty when predicting several stages into the future. Although a discount rate is usually defined as $0 < \gamma < 1$, for a finite horizon utility function as given by eq. (7.2) also $\gamma > 1$ is a legitimate choice.

The remainder of this section will be used to describe how the gradients of $\mathfrak{R}$ with respect to the control signals $a_t, \dots, a_{t+N-1}$ can be computed. This is technical but important because it will enable us to use gradient based optimisation methods to solve eq. (7.48). Let

$$\mathsf{R}_t(\boldsymbol{\mu}_t, \mathbf{S}_t, \mathbf{t}_t) \; = \; \mathop{\mathbb{E}}_{\mathbf{s_t}}[\mathsf{L}(\boldsymbol{s}_t, \mathbf{t}_t)] \; = \; \int_{\mathcal{S}} \mathsf{L}(\boldsymbol{s}_t, \mathbf{t}_t) \, \mathcal{N}(\boldsymbol{s}_t | \boldsymbol{\mu}_t, \mathbf{S}_t) \, d\boldsymbol{s}_t \tag{7.49}$$

denote the expected loss at time t where $\mathcal{N}(\boldsymbol{s}_t|\boldsymbol{\mu}_t, \mathbf{S}_t)$ is given by the dynamics GPs. Assuming $\mathsf{R}_t(\boldsymbol{\mu}_t, \mathbf{S}_t, \mathbf{t}_t)$ to be differentiable in $\boldsymbol{\mu}_t$ and $\mathbf{S}_t$, the gradient of the cumulated discounted loss (7.47) is given by

$$\frac{\partial \mathfrak{R}}{\partial a_{t+j}} = \sum_{i=1}^{N} \gamma^i \frac{\partial \mathsf{R}_{t+i}}{\partial a_{t+j}} = \sum_{i>j} \gamma^i \left(\frac{\partial \mathsf{R}_{t+i}}{\partial \boldsymbol{\mu}_{t+i}^\top} \frac{\boldsymbol{\mu}_{t+i}}{\partial a_{t+j}} + \frac{\partial \mathsf{R}_{t+i}}{\partial \mathbf{S}_{t+i}} \frac{\mathbf{S}_{t+i}}{\partial a_{t+j}} \right) \tag{7.50}$$

where $0 \leq j < N$.[3] We will compute this gradient for $i = j + 1$ and then propose an algorithm for iteratively computing the gradient over multiple time steps $i - j > 1$. By differentiating the mean (7.23), the variances (7.28), and the covariances (7.30) with respect to the parameters of the input distributions we can compute the terms

$$\frac{\partial \boldsymbol{\mu}_{t+1}}{\partial a_t}, \ \frac{\partial \boldsymbol{\mu}_{t+1}}{\partial \boldsymbol{s}_t}, \ \frac{\partial \boldsymbol{\mu}_{t+1}}{\partial \mathbf{S}_t}, \ \frac{\partial \mathbf{S}_{t+1}}{\partial a_t}, \ \frac{\partial \mathbf{S}_{t+1}}{\partial \boldsymbol{s}_t}, \ \frac{\partial \mathbf{S}_{t+1}}{\partial \mathbf{S}_t}, \tag{7.51}$$

which is all we need in order to compute the gradient of the expected cumulative loss $\mathfrak{R}$ with respect to all control signals $a_t, \ldots a_{t+N-1}$. When $i = j - 1$ we can directly compute

$$\frac{\partial \mathsf{R}_j}{\partial a_i} = \frac{\partial \mathsf{R}_j}{\partial \boldsymbol{\mu}_j^\top} \frac{\partial \boldsymbol{\mu}_j}{\partial a_i} + \frac{\partial \mathsf{R}_j}{\partial \mathbf{S}_j} \frac{\partial \mathbf{S}_j}{\partial a_i} \tag{7.52}$$

by simply plugging in the derivatives. But in case $i < j - 1$ we have to compute

$$\begin{aligned} \frac{\partial \mathsf{R}_j}{\partial a_i} = \frac{\partial \mathsf{R}_j}{\partial \boldsymbol{\mu}_j^\top} & \left(\frac{\partial \boldsymbol{\mu}_j}{\partial \boldsymbol{\mu}_{j-1}^\top} \frac{\partial \boldsymbol{\mu}_{j-1}}{\partial a_i} + \frac{\partial \boldsymbol{\mu}_j}{\partial \mathbf{S}_{j-1}} \frac{\partial \mathbf{S}_{j-1}}{\partial a_i} \right) \\ & + \frac{\partial \mathsf{R}_j}{\partial \mathbf{S}_j} \left(\frac{\partial \mathbf{S}_j}{\partial \boldsymbol{\mu}_{j-1}^\top} \frac{\partial \boldsymbol{\mu}_{j-1}}{\partial a_i} + \frac{\partial \mathbf{S}_j}{\partial \mathbf{S}_{j-1}} \frac{\partial \mathbf{S}_{j-1}}{\partial a_i} \right) \end{aligned} \tag{7.53}$$

for which we need to know the dependencies between actions and distributions over states for multiple time steps. Fortunately, we can state an algorithm in which we only need the first order derivatives (7.51). While iteratively predicting the N consecutive states we keep track of the partial derivatives of the respective state means and covariance matrices with respect to all previous control signals. In order to avoid further technicalities a schematic description of the gradient computation is given by Algorithm 8.

In the experiments presented below the optimal action is computed by minimising the discounted cumulated risk $\mathfrak{R}$ using a conjugate gradient based optimiser. In general, the optimisation problem (7.48) is not convex, so local minima problems can occur.

7.4.1. The Mountain Car Problem

The first experiment considers the mountain car problem which has already been described in Section 7.3.3. For model identification we generated 50 state transitions for

[3]Note that for notational simplicity we write $\frac{\partial a}{\partial \mathbf{A}} \frac{\partial \mathbf{A}}{\partial b}$ instead of $\frac{\partial a}{\mathrm{vec}(\mathbf{A})^\top} \frac{\mathrm{vec}(\mathbf{A})}{\partial b}$. Furthermore, terms of the form $\frac{\partial \mathbf{A}}{\partial \mathbf{B}}$ are handled like tensors and all operations are to be understood such that the chain rule holds.

Algorithm 8 Computation of the gradient of $\mathfrak{R}$ w.r.t. $a_t, \ldots a_{t+N-1}$.

```
**Given:** State $\boldsymbol{s}_0$, sequence of actions $a_0, \ldots, a_{N-1}$
Initialise $\boldsymbol{\mu}_0 = \boldsymbol{s}_0$, $\mathbf{S}_0 = \mathrm{diag}(\mathbf{0})$, $t \leftarrow 0$
**for** time step $i = 1, \ldots, N$ **do**
  Compute $\boldsymbol{\mu}_{t+1}$ and $\mathbf{S}_{t+1}$
  Compute partial derivatives of the predictive mean and covariance
      $$\frac{\partial \boldsymbol{\mu}_{t+1}}{\partial a_t}, \frac{\partial \boldsymbol{\mu}_{t+1}}{\partial \boldsymbol{\mu}_t^\top}, \frac{\partial \boldsymbol{\mu}_{t+1}}{\partial \mathbf{S}_t}, \frac{\partial \mathbf{S}_{t+1}}{\partial a_t}, \frac{\partial \mathbf{S}_{t+1}}{\partial \boldsymbol{\mu}_t}, \frac{\partial \mathbf{S}_{t+1}}{\partial \mathbf{S}_t}$$
  Compute and store partial derivative of the expected loss at time $t+1$
      $$\frac{\partial \mathsf{R}_{t+1}}{\partial a_t} = \frac{\partial \mathsf{R}_{t+1}}{\partial \boldsymbol{\mu}_{t+1}^\top}\frac{\partial \boldsymbol{\mu}_{t+1}}{\partial a_t} + \frac{\partial \mathsf{R}_{t+1}}{\partial \mathbf{S}_{t+1}}\frac{\partial \mathbf{S}_{t+1}}{\partial a_t}$$
  **for** all previous time steps $j = i-1, \ldots, 0$ **do**
    Update and store the partial derivatives
      $$\begin{aligned} \frac{\partial \boldsymbol{\mu}_{t+1}}{\partial a_j} &= \frac{\partial \boldsymbol{\mu}_{t+1}}{\partial \boldsymbol{\mu}_t^\top}\frac{\partial \boldsymbol{\mu}_t}{\partial a_j} + \frac{\partial \boldsymbol{\mu}_{t+1}}{\partial \mathbf{S}_t}\frac{\partial \mathbf{S}_t}{\partial a_j} \\ \frac{\partial \mathbf{S}_{t+1}}{\partial a_j} &= \frac{\partial \mathbf{S}_{t+1}}{\partial \boldsymbol{\mu}_t^\top}\frac{\partial \boldsymbol{\mu}_t}{\partial a_j} + \frac{\partial \mathbf{S}_{t+1}}{\partial \mathbf{S}_t}\frac{\partial \mathbf{S}_t}{\partial a_j} \\ \frac{\partial \mathsf{R}_{t+1}}{\partial a_j} &= \frac{\partial \mathsf{R}_{t+1}}{\boldsymbol{\mu}_{t+1}^\top}\frac{\boldsymbol{\mu}_{t+1}}{\partial a_j} + \frac{\partial \mathsf{R}_{t+1}}{\mathbf{S}_{t+1}}\frac{\mathbf{S}_{t+1}}{\partial a_j} \end{aligned}$$
  **end for**
  $t \leftarrow t+1$
**end for**
Finally compute the gradient vector element-wise
**for** $j = 0, \ldots, N-1$ **do**
      $$\frac{\partial \mathfrak{R}}{\partial a_{t+j}} = \sum_{j<i} \gamma^i \frac{\partial \mathsf{R}_{t+i}}{\partial a_{t+j}}$$
**end for**
Return gradient $\left(\frac{\partial \mathfrak{R}}{\partial a_0}, \ldots, \frac{\partial \mathfrak{R}}{\partial a_{N-1}}\right)^\top$
```

$\Delta t = 0.3$ to build the dynamics GPs. Note that $\Delta t = 0.3$ seconds seems to be an order of magnitude slower than the time scale usually considered in the literature.

Having a model of the system dynamics, the other necessary element of the proposed algorithm is a loss function. We choose the target state to be $\mathbf{t} = [0.6, 0]^\top$ and use a weighted quadratic loss function

$$\mathsf{L}(\mathbf{s}_t, \mathbf{t}) = (\boldsymbol{s}_t - \mathbf{t})^\top \begin{pmatrix} 1 & 0 \\ 0 & 0.1 \end{pmatrix} (\boldsymbol{s}_t - \mathbf{t}) \tag{7.54}$$

which emphasises that the main goal is to bring the car to the target x-position rather than not wanting the car to gain speed. Given a bivariate normal distribution $\mathcal{N}(\boldsymbol{s}_t|\boldsymbol{\mu}_t, \mathbf{S}_t)$ the expected loss at time t can be computed using the integral (B.26). In the experiment we used a five step horizon ($N = 5$) and thereby simulating the system behaviour over the next 1.5 seconds.

For the simulation illustrated in Figure 7.10 we initially placed the car at the deepest point of the valley $\boldsymbol{s}_0 = [-0.5, 0]^\top$. Then we applied control signals found by solving eq. (7.48) subject to the constraint $-4 \leq F_t \leq 4$. The strategy found is to first apply a force of -4 away from the target position to drive up the left hand slope from where again a force of 4 is used to speed up the car towards the target state utilising the landscape properties. Reaching the target state, the car is balanced around the target position with forces of approximately 2. Here it shows that bringing the car up to the target region can be solved using bang-bang like control strategies (only using maximum/minimum forces), but for balancing, more accurate forces have to be applied.

7.4.2. The Inverted Pendulum

We now turn to the inverted pendulum system that has been described in Section 7.3.4. For model identification we simulate 200 state transitions for $\Delta t = 0.2$ seconds and use ML-II to estimate the parameters of the dynamics GPs. The objective is to swing up the pendulum and balance it while the cart's position is at $x = 0$. Note that this is the task we could not solve using the approximate policy iteration algorithm in Section 7.3.4.

As loss function we use the squared distance

$$\mathsf{L}(\boldsymbol{s}) \;=\; d^2(\boldsymbol{s}) \;=\; (x + 2l\sin\theta)^2 + 4\,(l + l\cos\theta)^2 \tag{7.55}$$

between the pendulum tip in its actual position and its target position standing upright at $x = 0$, see again Figure 7.8. Note that the loss is a function of the position x and the angle θ only and is independent of the respective speeds.

We now approximate the expected loss R over the predictive distribution of the state $\mathcal{N}(\boldsymbol{s}|\boldsymbol{\mu}, \mathbf{S})$ given by the dynamics GPs. Therefore we use a second order Taylor series approximation of the loss function (7.55) around $\hat{\boldsymbol{s}}$:

$$d^2(\boldsymbol{s}) \;\approx\; d^2(\hat{\boldsymbol{s}}) + (\boldsymbol{s} - \hat{\boldsymbol{s}})^\top \nabla d^2(\hat{\boldsymbol{s}}) + \tfrac{1}{2}(\boldsymbol{s} - \hat{\boldsymbol{s}})^\top \nabla\nabla d^2(\hat{\boldsymbol{s}})(\boldsymbol{s} - \hat{\boldsymbol{s}}) \tag{7.56}$$

where the nonzero elements corresponding to x and θ of the gradient and the Hessian

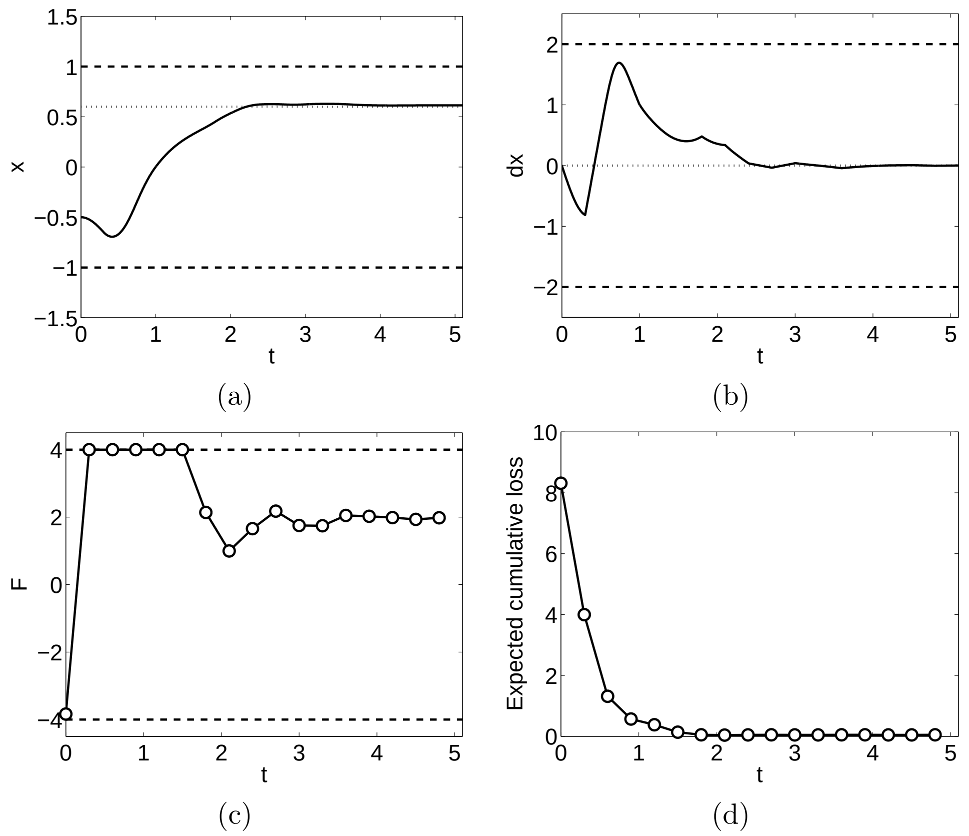


Figure 7.10.: Finite horizon planning for the mountain car problem. Panel (a) shows the x position and Panel (b) the speed $\dot{x}$ during a simulation of 5 seconds. Panel (c) shows the applied forces a_t. Note that the force has a negative sign, accelerating the car away from the target position. Panel (d) shows the value of the expected cumulative loss $\mathfrak{R}$.

are

$$\nabla d^2(\boldsymbol{s}) = \begin{pmatrix} 2x+4l\sin\theta \\ 4xl\cos\theta-8l^2\sin\theta \end{pmatrix} \quad \text{and} \quad \nabla\nabla d^2(\boldsymbol{s}) = \begin{pmatrix} 2 & 4l\cos\theta \\ 4l\cos\theta & -4xl\sin\theta-8l^2\cos\theta \end{pmatrix} .$$

When the Hessian becomes non-positive definite we add the absolute value of the smallest eigenvalue to the diagonal elements in order to ensure that the expected loss is positive. Let $\mathcal{N}(\boldsymbol{s}|\boldsymbol{\mu},\mathbf{S})$ and $\hat{\boldsymbol{s}} = \boldsymbol{\mu}$, then the expected loss can be approximated by taking the expectation over the Taylor approximation (7.56):

$$\mathsf{R}(\boldsymbol{s}) \;=\; \mathbb{E}[d^2(\boldsymbol{s})] \;\approx\; d^2(\boldsymbol{\mu}) + \tfrac{1}{2}\,\mathrm{tr}(\nabla\nabla d^2(\boldsymbol{\mu})\mathbf{S}) \tag{7.57}$$

using the integral (B.26).

Initially the cart stands at $x = 0$ with the pendulum hanging downwards ($\theta = 0$). For control we apply forces minimising the expected cumulative error $\mathfrak{R}$ over $N = 5$

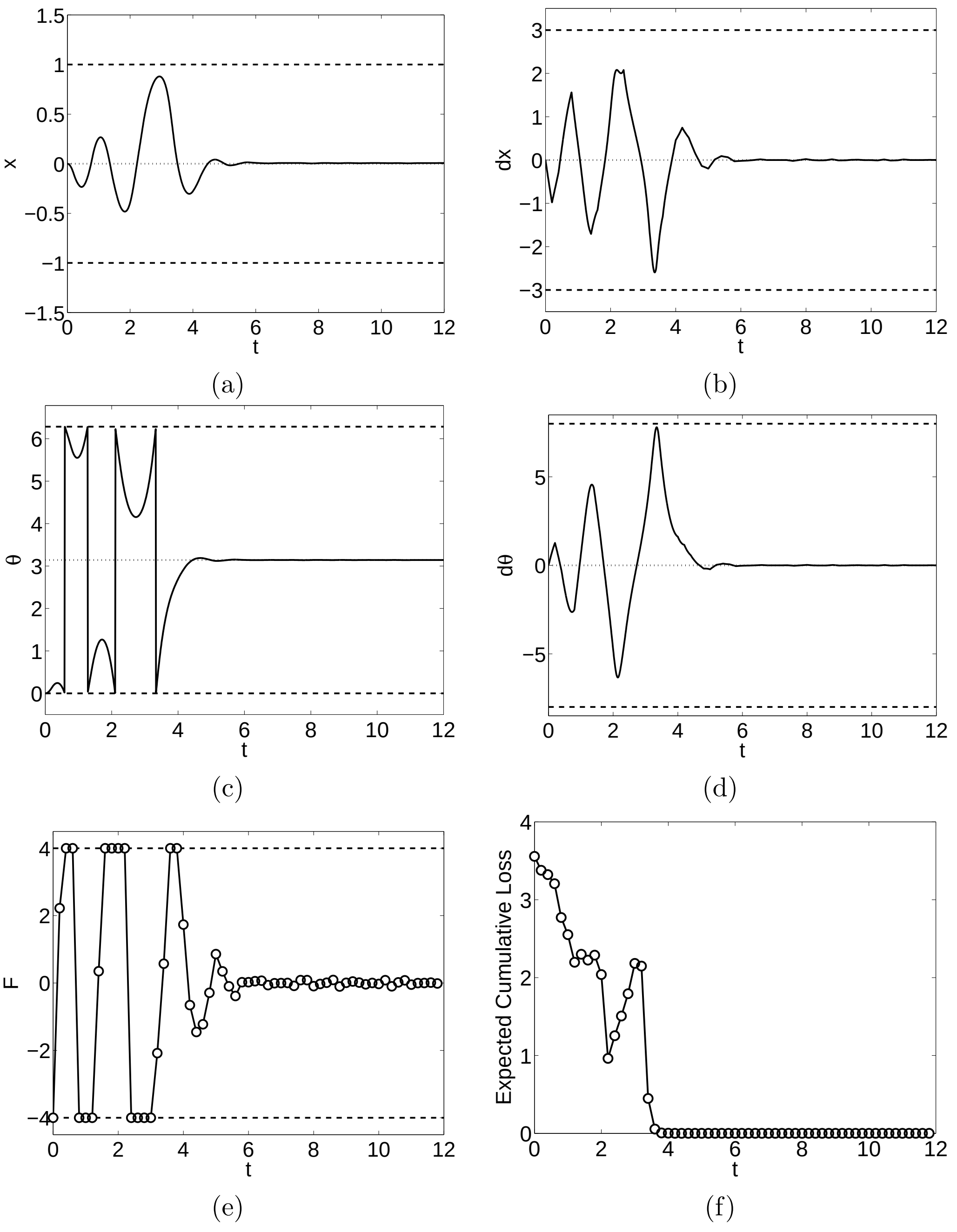


Figure 7.11.: Controlling the inverted pendulum system. Panels (a–d) display the state variables $\mathbf{s}_t = [x_t, \dot{x}_t, \theta_t, \dot{\theta}_t]^\top$ over time when applying control signals that minimise the expected cumulative loss over $N = 5$ time steps each $\Delta t = 0.2$sec in length. After about 4sec the pendulum is swung up and balanced for the remaining 8sec. The dashed lines mark the regions in which the training samples were generated from. Panel (e) shows the applied forces which were constrained to $-4 \leq F_t \leq 4$ (dashed lines) and Panel (f) shows the corresponding expected cumulative losses $\mathfrak{R}_t$ (discount factor $\gamma = 1$).

future time steps for $\gamma = 1$. We made the problem more difficult by constraining the control signal to be in the range $-4 \leq F \leq 4$. As shown in Figure 7.11, after about 4 seconds the pendulum can be brought into an upright position and is subsequently balanced with x positions close to the target value $x = 0$. For balancing the system it is evident from the system dynamics that the angle θ is by far the most sensitive variable. Hence the control policy has to focus on holding the pendulum upright, allowing only small deviations in θ while deviations in x are less critical. Note that $\Delta t = 0.2$sec is a rather long time interval for controlling the system which might be the reason for not being able to stabilise the pendulum completely. Local controllers may be better suited to balance the system around its optimum. We also sampled the state space around its optimum and found that a linear function is sufficient to balance the system well.

7.5. Conclusions & Discussion

In the course of this chapter several ideas were presented showing how Gaussian process models could be used in model-based reinforcement learning problems. The focus was how to implement planning in problems with continuous state spaces, as it appears in model based reinforcement learning. Gaussian process regression models have proven to be well suited to capture the non-linear dynamics of common benchmark problems based on a relatively small number of samples and relatively long time intervals.

In Section 7.3 we studied different approaches using a Gaussian process model to represent and estimate value functions. The approach constitutes an alternative to conventional function approximation methods based on temporal difference learning. Even if we are unable to implement Bayesian inference about the value functions—because it is unclear what the likelihood might be—the setting is interesting since it allows us to approximately solve a continuous equivalent to the Bellman equations. Despite this conceptual attractiveness, we have to emphasise that the proposed methods rely on the assumption that we can adequately capture the system dynamics and the value function using GP models given the set of support points. This implies potential difficulties in handling problems which exhibit discontinuous value functions, particularly in higher dimensions. Two drawbacks of GP models for representing the value function, either V or $\mathcal{Q}$, are the necessity to use a particular form of covariance function and the explicit representation of values at the support points.

In general it must be questioned whether a Gaussian process, especially with the stationary squared exponential covariance function (7.21), is well suited for representing the value function, which is often not stationary, i.e. its local properties vary over the state space. Another problem is that the Gaussian process model relies on a set of support points which need to lie dense enough to represent the value function accurately. If the state space is high dimensional, the placement and necessary number of support points becomes a difficult problem. Furthermore for each support point a non-convex optimisation problem has to be solved in order to update the corresponding estimate of the value.

Approximating value functions in continuous domains using non-linear function approximation, also using neural networks for instance, is always a difficult task. However, despite the problems mentioned above, the probabilistic approach studied in this chapter also comes with certain interesting aspects. Commonly the value function is defined to be the sum of *expected* (discounted) future rewards. Conceptually however, there is more to values than their expectations. The distribution over future reward could have small or large variance and identical means, two fairly different situations, that are treated identically when only the value expectation is considered. It is clear however, that a principled approach to the exploitation vs. exploration trade-off requires a more faithful representation of value, as was recently proposed in Bayesian Q-Learning for finite MDPs (Dearden et al., 1998). For example, the large variance case it may be more attractive for exploration than the small variance case.

The GP representation of value functions proposed here lends itself naturally to this more elaborate concept of value. The GP model represents a full distribution over values, although in the experiments above we have only used its expectation. Implementation of this would require a second set of Bellman-like equations for the second moment of the values at the support points. These equations would simply express consistency of uncertainty: the uncertainty of a value should be consistent with the uncertainty when following the policy.

Whereas only a batch version of the algorithm has been described, it would obviously be interesting to explore its capabilities in an online setting, starting from scratch. This will require that we abandon the use of a greedy policy, to avoid risking to get stuck in a local minima caused by an incomplete model of the dynamics. Instead, a stochastic policy should be used, which should not cause further computational problems as long as it is represented by a Gaussian (or perhaps more appropriately a mixture of Gaussians). A good policy should actively explore regions where we may gain a lot of information, requiring the notion of the value of information (Howard, 1966). Since the information gain would come from a better *dynamics* GP model, it may not be an easy task in practice to optimise jointly information and value.

We have demonstrated that the algorithm gives a reasonable approximation to the value function for the mountain car problem. However for the inverted pendulum system we could not succeed when using a reward function that included the location of the cart.

In practical problems it seems more relevant to find the optimal actions for particular states than it is to find the optimal strategy for all possible states. At first sight the finite horizon planning approach proposed in Section 7.4 might not appear very elegant, but it has shown to be able to swing-up and balance the inverted pendulum system only using observed examples of the dynamics. The approach could be used in control problems—maybe in conjunction with classical control techniques—where the differential equations describing the dynamics of the system are unknown, see Suykens et al. (2001) for a related approach. A major drawback of the proposed approach are computational costs related to the optimisations that are necessary at each time step. Although we derived the necessary gradients for the minimisation of the expected cumulative loss, the time necessary for optimisation in the Gaussian process models may take longer than the Δt interval such that online control would not be possible. The computations could be speed up for example by using sparse approximations to Gaussian process models as for example proposed by Lawrence et al. (2003). These sparse approximation could drastically reduce the computational cost for function evaluations as well as for computing the gradients. Another approach to do control online can be seen in pre-computing the control signal at relevant states and to make this accessible during actual control.

In summary, Gaussian process regression models come with certain attractive properties for model-based reinforcement learning. In this chapter several ideas have been studied for learning in MDPs with continuous state spaces in both the finite and infinite horizon settings. We have encountered practical problems that need to be worked on, but we have also demonstrated that the use of GP models opens new possibilities to study in the future.

7.6. Bibliographical Remarks

The term reinforcement learning has been introduced in the artificial intelligence community by Minsky (1961) and in control engineering by Waltz and Fu (1965). Very recommendable introductory texts on reinforcement learning were written by Kaelbling et al. (1996), Bertsekas and Tsitsiklis (1996), and Sutton and Barto (1998). The literature on Markov decision processes is large and many variations and extensions have been proposed. Recommendable introductions are to be found in White (1993) and Filar and Vrieze (1996). In the engineering literature the Markov decision process is referred to as controlled Markov chains (Kesten and Spitzer, 1975).

Sutton (1988) reviews previous uses of temporal difference learning and describes the TD(λ) algorithm. An early reference to the core concept of temporal difference learning is Samuel (1959). The convergence of temporal difference learning in finite MDPs has been proven by Dayan (1992) and Dayan and Sejnowski (1994). $\mathcal{Q}$-Learning has been developed by Watkins (1989), see also Watkins and Dayan (1992).

Bayesian approaches to the reinforcement learning problem are rare, mostly because the necessary computations are intractable and difficult to approximate. An early reference is the work of Martin (1967) in which Markov processes are approached from a Bayesian decision theoretic point of view. For finite MDPs the conjugate multinomial-Dirichlet model is proposed to learn the state transition probabilities, although it was named matrix beta distribution by the time. Strens (2000) proposed an interesting sampling approximation to the exploration vs. exploitation problem in finite MDPs, by sampling *worlds* from a probabilistic model of the environment. For many sampled *worlds* the optimal decision is computed using dynamic programming, which gives an approximate distribution over the optimal action.

Bandit processes are MDPs which have only one state and a choice of n possible actions (Berry and Fristedt, 1985). Gittins (1979, 1989) has shown that the optimal Bayesian strategy minimising the regret in Bandit problems can be computed, giving a non-approximate answer to the exploration vs. exploitation problem. Unfortunately, generalising this approach to MDPs with more than one state has shown to be very difficult (Duff, 2002).

Ormoneit and Sen (2002) proposed a kernel-based approach to reinforcement learning which asymptotically converges to a Gaussian process. Engel et al. (2003, 2005) proposed a Gaussian process based algorithm for model free reinforcement learning. The propagation of uncertainty in Gaussian process models for time series has been described by Girard et al. (2003) and Quiñonero-Candela et al. (2003). The idea of doing dynamic programming in continuous spaces using Gaussian process models has been described by Rasmussen and Kuss (2004).

A. Algorithms and Implementations

In fact, I ask you to remember only two points:

1. The tool that is so dull that you cannot cut yourself on it is not likely to be sharp enough to be either useful or helpful.
2. Most uses of the classical tools of statistics have been, are, and will be, made by those who know not what they do.

— Tukey (1965)

A.1. Implementing Gaussian Process Regression with Normal Noise

This appendix describes the implementation of Gaussian process regression with normal noise as described in Section 3.2. Hybrid Monte Carlo and ML-II estimation both require to evaluate the log evidence and to compute its gradient with respect to the hyper-parameters $\boldsymbol{\psi}$ and the likelihood parameters $\boldsymbol{\theta}$, which in this case refers to the noise variance $\sigma_{\mathfrak{n}}^2$ only. ML-II estimation is usually implemented by minimising the negative log evidence using a gradient based minimisation algorithm, e.g. a conjugate gradient method. In the regression setting with normal noise the negative log evidence comes in the form

$$-\ln p(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi}) \;=\; \tfrac{m}{2}\ln(2\pi) + \tfrac{1}{2}\ln\left|\mathbf{K}+\sigma_{\mathfrak{n}}^2\mathbf{I}\right| + \tfrac{1}{2}\mathbf{y}^\top(\mathbf{K}+\sigma_{\mathfrak{n}}^2\mathbf{I})^{-1}\mathbf{y} \tag{A.1}$$

where $\boldsymbol{\theta}=\sigma_{\mathfrak{n}}^2$. The optimisation algorithm in ML-II estimation or the Hybrid MCMC scheme repeatedly evaluates eq. (A.1) for various values of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. Each time the inverse and the determinant of $\mathbf{K}$ have to be re-computed which unfortunately scales cubic $\mathcal{O}(m^3)$ in time. A numerically stable approach to evaluating eq. (A.1) is to compute the Cholesky factorisation of $\mathbf{Q}=\mathbf{K}+\sigma_{\mathfrak{n}}^2\mathbf{I}$ such that $\mathbf{Q}=\mathbf{L}\mathbf{L}^\top$ (see Appendix B.1.5). Since $\mathbf{L}$ is triangular the inverse $\mathbf{L}^{-1}$ can be computed efficiently by back-substitution (Press et al., 2002, ch. 2.9). The log determinant of $\mathbf{Q}$ can be computed from $\mathbf{L}$ using relation (B.13).

Additionally the gradient of (A.1) has to be computed. By using the relations (B.10) one obtains (here for an element of $\boldsymbol{\psi}$ but likewise for $\sigma_{\mathfrak{n}}^2$):

$$\frac{-\partial \ln p(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})}{\partial\psi_i} \;=\; \frac{1}{2}\operatorname{tr}\left(\mathbf{Q}^{-1}\frac{\partial\mathbf{Q}}{\partial\psi_i}\right) - \frac{1}{2}\mathbf{y}^\top\mathbf{Q}^{-1}\frac{\partial\mathbf{Q}}{\partial\psi_i}\mathbf{Q}^{-1}\mathbf{y} \tag{A.2a}$$

$$\;=\; \frac{1}{2}\operatorname{tr}\left(\frac{\partial\mathbf{Q}}{\partial\psi_i}\left(\mathbf{Q}^{-1}-\mathbf{Q}^{-1}\mathbf{y}\mathbf{y}^\top\mathbf{Q}^{-1}\right)\right) \tag{A.2b}$$

where the second term in the trace is the same for all ψ_i. Note that for computing the trace of the product of two matrices the relation $\mathrm{tr}(\mathbf{AB}) = \sum_{i,j}(\mathbf{A} \odot \mathbf{B}^\top)$ can be used, where $\odot$ denotes the elementwise (Hadamard) product.

The predictive distribution for test cases is computed according to equations (3.12) where the inverse $\mathbf{L}^{-1}$ can be reused. Alternatively Gibbs and MacKay (1997) describe an iterative approximate procedure which is basically $\mathcal{O}(im^2)$ where i is the number of iterations.

A.2. Implementation of Laplace's Method

In Section 4.1 Laplace's method was described for approximate inference in Gaussian process models. This appendix describes details of our implementation, see also the appendices of Williams and Barber (1998).

Computing Laplace's approximation $\mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{A})$ for given $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ the main computational effort is involved in finding the maximum of the unnormalised log posterior $\ln \mathcal{Q}$ as given by eq. (4.2a). Finding the (or a local) mode can be implemented using a conjugate gradient optimisation scheme, but if the posterior is unimodal it is computationally advantageous to use Newton's method. In this case the vector $\mathbf{f}^0$ is initialised randomly and in each Newton step it is updated according to:

$$\begin{aligned} \mathbf{f}^{t+1} &= \mathbf{f}^t - (\nabla\nabla_{\mathbf{f}} \ln \mathcal{Q}(\mathbf{f}^t)^{-1} \nabla_{\mathbf{f}} \ln \mathcal{Q}(\mathbf{f}) && \text{(A.3a)} \\ &= (\mathbf{K}^{-1} + \mathbf{W})^{-1}(\mathbf{W}\mathbf{f}^t + \nabla_{\mathbf{f}} \ln \mathcal{L}(\mathbf{f}^t)) && \text{(A.3b)} \end{aligned}$$

until convergence of $\mathbf{f}$ to the mode $\mathbf{m}$. To ensure convergence the update is accepted only if the value of the target function increases, otherwise the step size is shortened until $\ln \mathcal{Q}(\mathbf{f}^{t+1}) \geq \ln \mathcal{Q}(\mathbf{f}^t)$.

Computationally Newton's method is dominated by the repeated inversion of the Hessian. Since $\mathbf{K}$ can be poorly conditioned we use the identity

$$(\mathbf{K}^{-1} + \mathbf{W})^{-1} = \mathbf{K} - \mathbf{K}\mathbf{W}^{\frac{1}{2}}(\mathbf{I} + \mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}})^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{K} \qquad \text{(A.4)}$$

such that only the well conditioned, positive definite matrix $(\mathbf{I}+\mathbf{W}^{1/2}\mathbf{K}\mathbf{W}^{1/2})$ has to be inverted. In our implementation the inverse is computed from a Cholesky decomposition of this matrix. If the likelihood is log-concave, $\mathbf{W}$ is a diagonal matrix with positive entries, hence computing $\mathbf{W}^{1/2}$ is trivial.

Note that implementing the Newton updates (A.3) only requires the *product* of the inverse Hessian times the gradient which could be computed more efficiently using an iterative conjugate gradient method, as for example described by Golub and Van Loan (1989, ch. 10).

Having found the mode $\mathbf{m}$ the approximation of the evidence (4.12) and its derivatives can be computed. The approximate evidence takes the form

$$\begin{aligned} \ln p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) &\approx \ln q(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi}) = \ln \mathcal{Q}(\mathbf{m}) + \tfrac{m}{2} \ln(2\pi) + \tfrac{1}{2} \ln |\mathbf{A}| && \text{(A.5a)} \\ &= \ln \mathcal{L}(\mathbf{m}) - \tfrac{1}{2}\mathbf{m}^\top \mathbf{K}^{-1}\mathbf{m} - \tfrac{1}{2} \ln |\mathbf{I} + \mathbf{K}\mathbf{W}| \ . && \text{(A.5b)} \end{aligned}$$

To avoid the direct inversion of $\mathbf{K}$ in the second term of (A.5b) we use the recurrence relation (A.3b). Let $\mathbf{a} = \mathbf{K}^{-1}\mathbf{m}$ then by substituting (A.4) into (A.3b) we obtain:

$$\mathbf{a} \;=\; (\mathbf{I} - \mathbf{W}^{\frac{1}{2}}(\mathbf{I} + \mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}})^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{K})(\mathbf{W}\mathbf{m} + \nabla_{\mathbf{f}} \ln \mathcal{L}(\mathbf{m})) \tag{A.6}$$

such that $\mathbf{m}^\top\mathbf{K}^{-1}\mathbf{m} = \mathbf{m}^\top\mathbf{a}$. The determinant in eq. (A.5b) can be rewritten

$$\ln|\mathbf{I} + \mathbf{K}\mathbf{W}| \;=\; \ln\left|\mathbf{I} + \mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}}\right| \tag{A.7}$$

and computed from the Cholesky decomposition, that was used to calculate the inverse in eq. (A.4). Note that if $\mathbf{M} = \mathbf{L}\mathbf{L}^\top$ is a Cholesky decomposition then $\ln|\mathbf{M}| = 2\sum \ln L_{ii}$, see Appendix B.1.5.

During ML-II estimation the approximate log evidence (A.5b) is maximised as a function of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. Our implementation is based on a conjugate gradient optimisation routine such that we also need to compute the derivatives of (A.5b). The dependency of the approximate evidence on $\boldsymbol{\psi}$ is two-fold:

$$\frac{\partial \ln q(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})}{\partial \psi_i} \;=\; \sum_{k,l} \frac{\partial \ln q(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})}{\partial K_{kl}} \frac{\partial K_{kl}}{\partial \psi_i} + \frac{\partial \ln q(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})}{\partial \mathbf{m}^\top} \frac{\partial \mathbf{m}}{\partial \psi_i}\,. \tag{A.8}$$

There is a direct dependency via the terms involving $\mathbf{K}$ and an implicit dependency through the effect on the mode $\mathbf{m}$ (see also Williams (1998, Appendix B)).

The explicit derivative of eq. (A.5b) due to the direct dependency of the covariance matrix is

$$\sum_{k,l} \frac{\partial \ln q(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})}{\partial K_{kl}} \frac{\partial K_{kl}}{\partial \psi_i} \;=\; \frac{1}{2}\mathbf{m}^\top\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \psi_i}\mathbf{K}^{-1}\mathbf{m} - \frac{1}{2}\operatorname{tr}\left((\mathbf{I} + \mathbf{K}\mathbf{W})^{-1}\frac{\partial \mathbf{K}}{\partial \psi_i}\mathbf{W}\right)$$

where the first term is computed using $\mathbf{a}$ (A.6) and the inverse in the second term can be rewritten as

$$(\mathbf{I} + \mathbf{K}\mathbf{W})^{-1} \;=\; \mathbf{I} - (\mathbf{K}^{-1} + \mathbf{W})^{-1}\mathbf{W} \tag{A.9}$$

where the inverse (A.4) is already known.

The implicit derivative accounts for the dependency of eq. (A.5b) on $\boldsymbol{\psi}$ due to change in the mode $\mathbf{m}$. Differentiating eq. (A.5a) with respect to $\mathbf{m}$ reduces to $\partial \ln|\mathbf{A}|/\partial \mathbf{m}$ since $\mathbf{m}$ is the maximum of $\ln \mathcal{Q}$ and therefore $\partial \ln \mathcal{Q}/\partial \mathbf{m}$ vanishes.

$$\frac{\partial \ln q(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})}{\partial \mathbf{m}^\top} \frac{\partial \mathbf{m}}{\partial \psi_i} \;=\; -\frac{1}{2}\frac{\partial|\mathbf{K}^{-1} + \mathbf{W}|}{\partial \mathbf{m}^\top}\frac{\partial \mathbf{m}}{\partial \psi_i} \tag{A.10a}$$

$$\;=\; -\frac{1}{2}(\mathbf{K}^{-1} + \mathbf{W})^{-1}\frac{\partial \mathbf{W}}{\partial \mathbf{m}^\top}\frac{\partial \mathbf{m}}{\partial \psi_i} \tag{A.10b}$$

The dependency of $\mathbf{m}$ on $\boldsymbol{\psi}$ is obtained by differentiating (4.8a) at $\mathbf{m}$:

$$0 = \nabla_{\mathbf{f}} \ln \mathcal{L}(\mathbf{m}) - \mathbf{K}^{-1}\mathbf{m} \quad\Longrightarrow\quad \mathbf{m} = \mathbf{K}\nabla_{\mathbf{f}} \ln \mathcal{L}(\mathbf{m}) \tag{A.11}$$

such that

$$\begin{aligned}\frac{\partial \mathbf{m}}{\partial \psi_i} &= \frac{\partial \mathbf{K}}{\partial \psi_i} \nabla_{\mathbf{f}} \ln \mathcal{L}(\mathbf{m}) + \mathbf{K}\, \nabla\nabla_{\mathbf{f}} \ln \mathcal{L}(\mathbf{m})\, \frac{\partial \mathbf{m}}{\partial \psi_i} && \text{(A.12)}\\ &= (\mathbf{I} + \mathbf{K}\mathbf{W})^{-1} \frac{\partial \mathbf{K}}{\partial \psi_i} \nabla_{\mathbf{f}} \ln \mathcal{L}(\mathbf{m}) && \text{(A.13)}\end{aligned}$$

and we have both terms necessary to compute the gradient (A.8).

For computing the derivatives of the approximate evidence (A.5b) with respect to $\boldsymbol{\theta}$ one has to account for a direct dependency through $\ln \mathcal{L}$ and $\mathbf{W}$ and again an indirect dependency through the effect on the mode

$$\frac{\partial q(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi})}{\partial \theta_i} = \frac{\partial \ln \mathcal{L}}{\partial \theta_i} - \frac{\partial \ln |\mathbf{I} + \mathbf{K}\mathbf{W}|}{2\partial \theta_i} + \frac{\partial q(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi})}{\partial \mathbf{m}^\top} \frac{\partial \mathbf{m}}{\partial \theta_i} \qquad \text{(A.14)}$$

where the last term is obtained by differentiating the right hand side of eq. (A.11).

The approximate posterior Gaussian process (4.3) has mean and covariance function

$$\begin{aligned} m_*(\mathbf{x}) &= \mathbf{k}(\mathbf{x})^\top \mathbf{K}^{-1}\mathbf{m} = \mathbf{k}(\mathbf{x})^\top \mathbf{a} && \text{(A.15a)}\\ k_*(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x})^\top \mathbf{W}^{\frac{1}{2}} (\mathbf{I} + \mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}})^{-1} \mathbf{W}^{\frac{1}{2}} \mathbf{k}(\mathbf{x}') && \text{(A.15b)}\end{aligned}$$

where $\mathbf{a}$ is given by eq. (A.6) and the inverse (A.4) can be reused.

Algorithm 9 Laplace's approximation using Newton's method for GP models

```
**Given:** $\boldsymbol{\theta}$, $\boldsymbol{\psi}$, $\mathcal{D}$, $\mathbf{x}_*$
Initialise $\mathbf{f}$ (e.g. $\mathbf{f} \leftarrow \mathbf{0}$ or randomly), compute $\mathbf{K}$ from $\boldsymbol{\psi}$ and $\mathbf{X}$
**repeat**
  $\mathbf{f} \leftarrow \mathbf{f} - (\nabla\nabla_{\mathbf{f}} \ln \mathcal{Q}(\mathbf{f}))^{-1} \nabla_{\mathbf{f}} \ln \mathcal{Q}(\mathbf{f})$
**until** convergence of $\mathbf{f}$
$\mathbf{m} \leftarrow \mathbf{f}$
$\mathbf{A} \leftarrow (\mathbf{K}^{-1} - \nabla\nabla_{\mathbf{f}} \ln \mathcal{Q}(\mathbf{m}))^{-1}$
Compute approximate log evidence $\ln q(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi})$ by (A.5), and predictions
$q(f_*|\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{x}_*)$ using (A.15).
```

Since each evaluation of $\ln \mathcal{Q}$ as given by eq. (4.2a) requires a Cholesky decomposition of the covariance matrix, computing Laplace's approximation is cubic $\mathcal{O}(m^3)$ in time. However, following the implementation described in this section, the Cholesky decomposition has to be computed once per Newton step and all other quantities can be computed from it in at most $\mathcal{O}(m^2)$. The number of Newton steps necessary depends on the convergence criterion, the initialisation of $\mathbf{f}$, the values of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$, etc., but in our experiments the algorithm usually converged after a few (typically less than ten) iterations. Finding the mode using a conjugate gradient method requires significantly more evaluations of eq. (4.2a) and is therefore unfavourable. If the posterior is not unimodal we typically used a conjugate gradient method to find a local mode and repeated the procedure for several initial values.

A.3. Implementation of Expectation Propagation

In this appendix we describe details of our implementation of Expectation Propagation as described in Section 4.2 and summarised in Algorithm 10. See also the appendices of Seeger (2003) and Kuss and Rasmussen (2005), the latter especially for the probit GP model for binary classification described in Chapter 6.

In our implementation the site functions (4.14) are parameterised in terms of natural parameters σ_i^{-2} and $\sigma_i^{-2}\mu_i$ (Schervish, 1997, ch. 2.2). For given $\boldsymbol{\psi}$ the algorithm starts by initialising $\mathbf{A} = \mathbf{K}$, $\sigma_i^{-2} = 0$, and $\sigma_i^{-2}\mu_i = 0$ for all $i = 1, \ldots, m$. The algorithm proceeds by updating the site parameters in random order. In each sweep every site function is updated following equations (4.20) and (4.25). After each update of a site function the effect on $\mathbf{m}$ and $\mathbf{A}$ has to be computed according to eq. (4.15). The change of $\mathbf{A}$ can be computed using a rank-one update. Let δ be the change in σ_i^{-2} due to the update and $\mathbf{e}_i$ the $[m \times 1]$ vector whose ith entry is 1 and all other 0. The relation

$$(\mathbf{K}^{-1} + \boldsymbol{\Sigma}^{-1} + \delta \mathbf{e}_i \mathbf{e}_i^\top)^{-1} \;=\; \mathbf{A} - \mathbf{A}\mathbf{e}_i(\mathbf{A}_{ii} + \delta^{-1})^{-1}\mathbf{e}_i^\top \mathbf{A} \tag{A.16}$$

can be used to update $\mathbf{A}$. Each single update is $\mathcal{O}(m^2)$ and repeated m times per sweep, such that the EP algorithm scales $\mathcal{O}(m^3)$ in time. Because of accumulating numerical errors, after a complete sweep over all site functions we recompute the matrix $\mathbf{A}$ from scratch. For numerical stability we rewrite

$$\mathbf{A} = (\mathbf{K}^{-1} + \boldsymbol{\Sigma}^{-1})^{-1} \;=\; \mathbf{K} - \mathbf{K}\boldsymbol{\Sigma}^{-\frac{1}{2}}(\mathbf{I} + \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{K}\boldsymbol{\Sigma}^{-\frac{1}{2}})^{-1}\boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{K} \tag{A.17}$$

and compute the inverse from the Cholesky decomposition of $(\mathbf{I} + \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{K}\boldsymbol{\Sigma}^{-\frac{1}{2}})$.

Algorithm 10 EP approximation for Gaussian process models

```
Given: $\boldsymbol{\theta}$, $\boldsymbol{\psi}$, $\mathcal{D}$, $\mathbf{x}_*$
Initialise: $\mathbf{A} \leftarrow \mathbf{K}$ and site parameters $\sigma_i^2$ and $\mu_i$
repeat
  for i=1,...,m do
    Compute parameters (4.20) of approximate cavity
    Compute moments according to the likelihood model
    Update the site parameters using (4.25)
    Update $\mathbf{m}$ and $\mathbf{A}$ according to (4.15)
  end for
until The site parameters converged
Compute approximate log evidence $\ln q(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})$ by (4.26), and predictions
$q(f_*|\mathcal{D},\boldsymbol{\theta},\boldsymbol{\psi},\mathbf{x}_*)$ using (A.21).
```

After convergence the approximate log evidence (4.26) can be computed and the partial derivatives with respect to the hyper-parameters are:

$$\frac{\partial \ln q(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\psi})}{\partial \psi_i} \;=\; -\frac{1}{2}\,\mathrm{tr}\left(\frac{\partial \mathbf{K}}{\partial \psi_i}\left((\mathbf{K}+\boldsymbol{\Sigma})^{-1} - (\mathbf{K}+\boldsymbol{\Sigma})^{-1}\boldsymbol{\mu}\boldsymbol{\mu}^\top(\mathbf{K}+\boldsymbol{\Sigma})^{-1}\right)\right)\,.$$

The inverse of $\mathbf{K} + \mathbf{\Sigma}$ can be computed from the inverse in eq. (A.17):

$$(\mathbf{K} + \mathbf{\Sigma})^{-1} \;=\; \mathbf{\Sigma}^{-\frac{1}{2}}(\mathbf{I} + \mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{K}\mathbf{\Sigma}^{-\frac{1}{2}})^{-1}\mathbf{\Sigma}^{-\frac{1}{2}}\;. \tag{A.18}$$

For computing the log evidence (4.26) also the determinant $|\mathbf{K}+\mathbf{\Sigma}|$ has to be computed. By rewriting

$$\ln|\mathbf{K} + \mathbf{\Sigma}| \;=\; \ln(|\mathbf{\Sigma}||\mathbf{I} + \mathbf{\Sigma}^{-1}\mathbf{K}|) \;=\; \ln|\mathbf{\Sigma}| + \ln|\mathbf{I} + \mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{K}\mathbf{\Sigma}^{-\frac{1}{2}}| \tag{A.19}$$

we obtain an expression in which the first term is a determinant of a diagonal matrix and the second term can be computed from the Cholesky decomposition that was used to compute the inverse in eq. (A.17).

The derivative of the approximate log evidence (4.26) with respect to the likelihood parameters $\boldsymbol{\theta}$ is simply

$$\frac{\partial \ln q(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi})}{\partial \theta_i} \;=\; \sum_{j=1}^{m} \frac{\partial \ln Z_j}{\partial \theta_i} \;=\; \sum_{j=1}^{m} \frac{\partial \ln m_0^j}{\partial \theta_i} \tag{A.20}$$

where m_0^j is the zeroth moment (4.24) corresponding to the jth observation.

To compute the mean and covariance function of the GP approximation to the posterior process we have:

$$m_*(\mathbf{x}) \;=\; \mathbf{k}(\mathbf{x})^\top(\mathbf{K} + \mathbf{\Sigma})^{-1}\boldsymbol{\mu} \tag{A.21a}$$

$$k_*(\mathbf{x}, \mathbf{x}') \;=\; k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x})^\top(\mathbf{K} + \mathbf{\Sigma})^{-1}\mathbf{k}(\mathbf{x}')\,. \tag{A.21b}$$

where we can reuse the inverse (A.18).

Like Laplace's method, the EP algorithm is of computational complexity $\mathcal{O}(m^3)$ due to the computations for updating $\mathbf{A}$. However, per sweep the computation of $\mathbf{A}$ (A.17) and the m rank-one updates typically result in longer running times compared to Laplace's method.

A.4. Implementation of MCMC Sampling

A.4.1. Hybrid Monte Carlo

The Hybrid Monte Carlo method of Duane et al. (1987) has been described in Section 4.3.4. This appendix provides pseudo-code for implementing the algorithm, see also MacKay (2003, ch. 30) (in which the method is called *Hamiltonian* Monte Carlo). An implementation for the R environment for statistical computing is contained in the "PsychoFun" package accompanying Kuss et al. (2005a).

Algorithm 11 Hybrid MCMC Sampling

```
Given: Initial state $\boldsymbol{\phi}_0$, length of simulation $T$, number of leapfrog steps $l$, vector of
leapfrog step sizes $\boldsymbol{\epsilon}$, potential energy function $E(\boldsymbol{\phi}) = -\ln \mathcal{Q}(\boldsymbol{\phi}|\mathcal{D})$
Initialise $e \leftarrow E(\boldsymbol{\phi}_0)$ and $\mathbf{g} \leftarrow \nabla E(\boldsymbol{\phi}_0)$
for $t = 0, \dots, T$ do
  Sample initial momentum $\mathbf{q}$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$
  $H \leftarrow \frac{\mathbf{q}^\top \mathbf{q}}{2} + e$
  Set $\tilde{\boldsymbol{\phi}} \leftarrow \boldsymbol{\phi}_t$ and $\tilde{\mathbf{g}} \leftarrow \mathbf{g}$
  for all leapfrog steps do
    $\mathbf{q} \leftarrow \mathbf{q} - \frac{1}{2}\boldsymbol{\epsilon} \odot \tilde{\mathbf{g}}$ {where $\odot$ denotes the element-wise product}
    $\tilde{\boldsymbol{\phi}} \leftarrow \tilde{\boldsymbol{\phi}} + \boldsymbol{\epsilon} \odot \mathbf{q}$
    $\tilde{\mathbf{g}} \leftarrow \nabla E(\tilde{\boldsymbol{\phi}})$
    $\mathbf{q} \leftarrow \mathbf{q} - \frac{1}{2}\boldsymbol{\epsilon} \odot \tilde{\mathbf{g}}$
  end for
  $\tilde{e} \leftarrow E(\tilde{\boldsymbol{\phi}})$
  $\tilde{H} \leftarrow \frac{\mathbf{q}^\top \mathbf{q}}{2} + \tilde{e}$
  Draw $u$ from a uniform distribution on $[0, 1]$
  if $\ln u > \tilde{H} - H$ then {Proposed state is accepted}
    $\boldsymbol{\phi}_{t+1} \leftarrow \tilde{\boldsymbol{\phi}}$
    $\mathbf{g} \leftarrow \tilde{\mathbf{g}}$
    $e \leftarrow E(\boldsymbol{\phi}_{t+1})$
  else {Proposed state is rejected}
    $\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t$
  end if
  Store $\boldsymbol{\phi}_{t+1}$ and $E(\boldsymbol{\phi}_{t+1})$
end for
Return: $\boldsymbol{\phi}_t$ and $E(\boldsymbol{\phi}_t)$ for $t = 1, \dots, T$
```

A.4.2. Annealed Importance Sampling

Section 4.3.5 described Annealed Importance Sampling for estimating the log evidence $p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\psi})$ in Gaussian process models. Note that in principle one could also integrate over the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. Algorithm 12 describes the implementation that was used for the example in Section 4.3.6.2 and the experiments in Section 6.3.

Algorithm 12 Annealed Importance Sampling for Gaussian process models

```
Given: Temperature schedule $\tau$
for $r = 1, \ldots, R$ do
    Sample $\mathbf{f}_0$ from the prior $\mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$

    for $t = 1, \ldots, T$ do
        Sample $\mathbf{f}_t$ from $q(\mathbf{f}|\mathcal{D}, \tau(t), \boldsymbol{\theta}, \boldsymbol{\psi})$ as given by eq. (4.43) using Hybrid Monte Carlo
        (initial state is set to $\mathbf{f}_{t-1}$)
        Compute $\ln(Z_t/Z_{t-1})$ using (4.45)
    end for

    Compute approximate $Z_r$ using (4.46)
end for
Return $\ln Z = \ln\left(\frac{1}{R}\sum_{r=1}^{R} Z_r\right)$
```

B. Mathematical Appendix

When you choose how much postage to use,
When you know what's the chance it will snow,
When you bet and you end up in debt,
Oh try as you may, you just can't get away
From mathematics! — Tom Lehrer

B.1. Matrix Analysis

The stated results can be found in Lütkepohl (1996), Harville (1997), Golub and Van Loan (1989), or Press et al. (2002).

B.1.1. Identities Involving Inverses

Let $\mathbf{A}$ and $\mathbf{B}$ be non-singular square $[m \times m]$ matrices. The inverse of the product of the two matrices can be written in terms of the individual inverses

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \tag{B.1}$$

and for the product with a scalar c holds $(c\mathbf{A})^{-1} = c^{-1}\mathbf{A}^{-1}$. For the sum of two matrices the following identities are valid:

$$\mathbf{A}^{-1} + \mathbf{B}^{-1} = \mathbf{A}^{-1}(\mathbf{A}+\mathbf{B})\mathbf{B}^{-1} \tag{B.2}$$

$$(\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1} = \mathbf{A}(\mathbf{A}+\mathbf{B})^{-1}\mathbf{B} = \mathbf{B}(\mathbf{A}+\mathbf{B})^{-1}\mathbf{A}\ . \tag{B.3}$$

The matrix inversion lemma, also known as the Sherman-Morrison-Woodbury formula states the identity:

$$(\mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{D})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{C}(\mathbf{B} + \mathbf{D}\mathbf{A}^{-1}\mathbf{C})^{-1}\mathbf{D}\mathbf{A}^{-1} \tag{B.4}$$

if all the inverses exit. The inverse of a partitioned matrix can be described in terms of its sub-matrices

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1}+\mathbf{A}^{-1}\mathbf{B}(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & (\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix} \tag{B.5}$$

if the inverses $\mathbf{A}^{-1}$ and $(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}$ exist.

B.1.2. The Pseudoinverse

Let $\mathbf{A}$ be an $[m \times m']$ matrix of rank $r \leq \min(m, m')$. The generalised inverse is an extension of the concept of a matrix inverse to singular and general rectangular matrices. A generalised inverse $\mathbf{A}^-$ of $\mathbf{A}$ is any $[m' \times m]$ matrix that holds

$$\mathbf{A}\mathbf{A}^-\mathbf{A} = \mathbf{A} \tag{B.6}$$

The pseudo-inverse, also known as Moore-Penrose inverse, $\mathbf{A}^+$ is a unique generalised inverse that has the additional properties

$$\begin{aligned}
\mathbf{A}\mathbf{A}^+\mathbf{A} &= \mathbf{A} && \text{(B.7a)}\\
\mathbf{A}^+\mathbf{A}\mathbf{A}^+ &= \mathbf{A}^+ && \text{(B.7b)}\\
(\mathbf{A}^+\mathbf{A})^\top &= \mathbf{A}^+\mathbf{A} && \text{(B.7c)}\\
(\mathbf{A}\mathbf{A}^+)^\top &= (\mathbf{A}\mathbf{A}^+) && \text{(B.7d)}
\end{aligned}$$

Penrose conditions (Penrose, 1955). If $\mathbf{A}$ is square and non-singular, the pseudo-inverse coincides with the common inverse $\mathbf{A}^+ = \mathbf{A}^{-1}$. If $\mathbf{A}$ is square, symmetric, but singular with rank $r < m$, the generalised inverse can be computed from the eigen-decomposition $\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ as $\mathbf{A}^+ = \mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U}^\top$ where $\boldsymbol{\Lambda}$ is a diagonal $[r \times r]$ matrix of nonzero eigenvalues.

B.1.3. Identities Involving Determinants

Let $\mathbf{A}$ and $\mathbf{B}$ be arbitrary $[m \times m]$ matrices. The following identities are helpful:

$$\begin{aligned}
|\mathbf{A}| &= \left|\mathbf{A}^\top\right| && \text{(B.8a)}\\
|c\mathbf{A}| &= c^n\,|\mathbf{A}| && \text{(B.8b)}\\
|\mathbf{A}|^c &= |\mathbf{A}^c| && \text{(B.8c)}\\
|\mathbf{A}\mathbf{B}| &= |\mathbf{A}|\,|\mathbf{B}| && \text{(B.8d)}
\end{aligned}$$

B.1.4. Matrix Derivatives

The derivative of a matrix $\mathbf{A}$ with respect to a scalar $\partial\mathbf{A}/\partial x$ is defined element-wise such that

$$\left(\frac{\partial \mathbf{A}}{\partial x}\right)_{ij} = \frac{\partial A_{ij}}{\partial x}\,. \tag{B.9}$$

When differentiating traces, inverses and determinants the following relations are helpful:

$$
\begin{aligned}
\frac{\partial \operatorname{tr}(\mathbf{A})}{\partial x} &= \operatorname{tr}\left(\frac{\partial \mathbf{A}}{\partial x}\right) && \text{(B.10a)}\\
\frac{\partial \mathbf{A}^{-1}}{\partial x} &= -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\mathbf{A}^{-1} && \text{(B.10b)}\\
\frac{\partial |\mathbf{A}|}{\partial x} &= |\mathbf{A}| \operatorname{tr}\left(\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\right) && \text{(B.10c)}\\
\frac{\partial \ln |\mathbf{A}|}{\partial x} &= \operatorname{tr}\left(\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\right) && \text{(B.10d)}
\end{aligned}
$$

For differentiating a scalar valued function with respect to a matrix argument:

$$\frac{\partial \operatorname{tr}(\mathbf{AB})}{\mathbf{A}} = \mathbf{B}^\top \quad \text{and} \quad \frac{\partial \ln |\mathbf{A}|}{\mathbf{A}} = (\mathbf{A}^\top)^{-1}\,. \tag{B.11}$$

For more details on matrix differential calculus see Lütkepohl (1996, ch. 10), Harville (1997, ch. 15), and Magnus and Neudecker (1988).

B.1.5. The Cholesky Decomposition

Let $\mathbf{A}$ be an $[m \times m]$ symmetric, strictly positive definite matrix, then $\mathbf{A}$ can be decomposed as

$$\mathbf{A} = \mathbf{L}\mathbf{L}^\top \tag{B.12}$$

where the so called Cholesky factor $\mathbf{L}$ is a lower triangular matrix . The Cholesky factorisation of $\mathbf{A}$ can be used to compute the inverse of $\mathbf{A}^{-1}$ and its determinant $|\mathbf{A}|$. The inverse is $\mathbf{A}^{-1} = (\mathbf{L}^{-1})^\top \mathbf{L}^{-1}$ where $\mathbf{L}^{-1}$ can be computed in time quadratic in m due to the triangular structure by back-substitution (Press et al., 2002, ch. 2.9). The determinant and the log determinant can be computed according to

$$|\mathbf{A}| = \prod_{i=1}^{m} L_{ii}^2 \quad \text{and} \quad \ln |\mathbf{A}| = 2\sum_{i=1}^{m} \ln L_{ii} \tag{B.13}$$

which is linear in time.

B.1.6. Quadratic Forms

When handling products of Gaussian variables the following relation is frequently used

$$
\begin{aligned}
&(\mathbf{a}-\mathbf{b})^\top \mathbf{A}^{-1}(\mathbf{a}-\mathbf{b}) + (\mathbf{a}-\mathbf{c})^\top \mathbf{B}^{-1}(\mathbf{a}-\mathbf{c})\\
&\qquad = (\mathbf{a}-\mathbf{z})^\top(\mathbf{A}^{-1}+\mathbf{B}^{-1})(\mathbf{a}-\mathbf{z}) + (\mathbf{b}-\mathbf{c})^\top(\mathbf{A}+\mathbf{B})^{-1}(\mathbf{b}-\mathbf{c}) \qquad \text{(B.14)}
\end{aligned}
$$

where $\mathbf{z} = (\mathbf{A}^{-1}+\mathbf{B}^{-1})^{-1}(\mathbf{A}^{-1}\mathbf{b}+\mathbf{B}^{-1}\mathbf{c})$. See Box and Tiao (1973, Appendix A7.1) for a proof.

B.2. The Multivariate Normal Distribution

The multivariate normal distribution is covered by almost all textbooks on multivariate statistics and probability theory. A recommendable summary is given by Mardia et al. (1979, ch. 3).

B.2.1. Probability Density Function

Let $\mathbf{x}$ be an m-dimensional multivariate normal random variable, then its probability density function is given by

$$p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \tag{B.15a}$$
$$= (2\pi)^{-m/2}\,|\boldsymbol{\Sigma}|^{-1/2}\exp\left(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{B.15b}$$

where $\boldsymbol{\mu}\in\mathbb{R}^m$ denotes the mean and $\boldsymbol{\Sigma}$ is a symmetric, positive definite $[m\times m]$ covariance matrix.

B.2.2. Gaussian Identities

B.2.2.1. Products of Two Gaussians

Let $\mathbf{x}$, $\mathbf{a}$ and $\mathbf{b}$ be of size $[m\times 1]$ and $\mathbf{A}$ and $\mathbf{B}$ be $[m\times m]$ covariance matrices. The product of two multivariate normal distributions is proportional to another multivariate normal distribution

$$\mathcal{N}(\mathbf{x}|\mathbf{a},\mathbf{A})\,\mathcal{N}(\mathbf{x}|\mathbf{b},\mathbf{B}) = Z\,\mathcal{N}(\mathbf{x}|\mathbf{c},\mathbf{C}) \tag{B.16}$$

with covariance and mean

$$\mathbf{C} = \left(\mathbf{A}^{-1}+\mathbf{B}^{-1}\right)^{-1} \quad\text{and}\quad \mathbf{c} = \mathbf{C}\left(\mathbf{A}^{-1}\mathbf{a}+\mathbf{B}^{-1}\mathbf{b}\right)\,. \tag{B.17}$$

The normalising constant Z is Gaussian in either $\mathbf{a}$ or $\mathbf{b}$

$$z_c = (2\pi)^{-\frac{m}{2}}\left|\mathbf{A}\mathbf{B}\mathbf{C}^{-1}\right|^{-\frac{1}{2}}\exp\left(-\tfrac{1}{2}\left(\mathbf{a}^\top\mathbf{A}^{-1}\mathbf{a}+\mathbf{b}^\top\mathbf{B}^{-1}\mathbf{b}-\mathbf{c}^\top\mathbf{C}^{-1}\mathbf{c}\right)\right) \tag{B.18}$$

Let $\mathbf{y}$ be an $[m'\times 1]$ Gaussian random variable whose mean depends linearly on $\mathbf{x}$ where $\mathbf{D}$ is of size $[m'\times m]$, and $\mathbf{B}$ is $[m'\times m']$. Then the product

$$\mathcal{N}(\mathbf{x}|\mathbf{a},\mathbf{A})\,\mathcal{N}(\mathbf{y}|\mathbf{D}\mathbf{x},\mathbf{B}) \propto \mathcal{N}(\mathbf{x}|\mathbf{c},\mathbf{C}) \tag{B.19}$$

is again proportional to a multivariate normal density with mean and covariance

$$\mathbf{c} = \mathbf{C}\left(\mathbf{A}^{-1}\mathbf{a}+\mathbf{D}^\top\mathbf{B}^{-1}\mathbf{y}\right) \quad\text{and}\quad \mathbf{C} = \left(\mathbf{A}^{-1}+\mathbf{D}^\top\mathbf{B}^{-1}\mathbf{D}\right)^{-1}\,. \tag{B.20}$$

B.2.2.2. Marginal and Conditional Distributions

Let $\mathbf{x} \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be partitioned $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]^\top$ such that

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \middle| \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right) \tag{B.21}$$

then the marginal distributions are $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$ and $\mathbf{x}_2 \sim \mathcal{N}(\mathbf{x}_2|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22})$. The conditional distributions are:

$$\mathbf{x}_1|\mathbf{x}_2 \;\sim\; \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}) \tag{B.22a}$$

$$\mathbf{x}_2|\mathbf{x}_1 \;\sim\; \mathcal{N}(\mathbf{x}_2|\boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\mathbf{x}_1 - \boldsymbol{\mu}_1), \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12})\;. \tag{B.22b}$$

B.2.2.3. Linear Forms

Let $\mathbf{x} \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{c}$ then $\mathbf{y} \sim \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{c}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top)$.

B.2.3. Gaussian Integrals

By definition the probability density function integrates to one

$$\int_{\mathbb{R}^m} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x} \;=\; 1\,. \tag{B.23}$$

B.2.3.1. Two Gaussians

$$\int_{\mathbb{R}^m} \mathcal{N}(\mathbf{x}|\mathbf{a}, \mathbf{A})\, \mathcal{N}(\mathbf{a}|\mathbf{b}, \mathbf{B})\, d\mathbf{a} \;=\; \mathcal{N}(\mathbf{x}|\mathbf{b}, \mathbf{A} + \mathbf{B}) \tag{B.24}$$

B.2.3.2. Linear forms

$$\int_{\mathbb{R}^n} (\mathbf{c}^\top\mathbf{x})\, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x} \;=\; \mathbf{c}^\top\boldsymbol{\mu} \tag{B.25}$$

B.2.3.3. Quadratic forms

$$\int_{\mathbb{R}^n} (\mathbf{x} - \mathbf{c})^\top\mathbf{A}(\mathbf{x} - \mathbf{c})\, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x} \;=\; (\boldsymbol{\mu} - \mathbf{c})^\top\mathbf{A}(\boldsymbol{\mu} - \mathbf{c}) + \mathrm{tr}(\mathbf{A}\boldsymbol{\Sigma}) \tag{B.26}$$

B.2.3.4. Outer products

$$\int_{\mathbb{R}^n} \mathbf{x}\,\mathbf{x}^\top\, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x} \;=\; \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top \tag{B.27}$$

B.2.3.5. Entropy

The entropy of a probability distribution $p(\mathbf{x})$ is defined as

$$\mathcal{H}(p) \;=\; -\int p(\mathbf{x})\, \log p(\mathbf{x})\, d\mathbf{x} \tag{B.28}$$

see for example (Cover and Thomas, 1991, ch. 2.1). If the logarithm is taken to the base 2, the entropy is said to be measured in *bits* and in *nats* if base e is used. The entropy of a Gaussian random variable $\mathbf{x} \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ in nats is:

$$\mathcal{H}_{\mathcal{N}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\int \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \ln \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x} \tag{B.29a}$$

$$= \tfrac{m}{2}\ln(2\pi) + \tfrac{1}{2}\ln|\boldsymbol{\Sigma}| + \tfrac{m}{2} = \tfrac{1}{2}\ln|2\pi e \boldsymbol{\Sigma}| \tag{B.29b}$$

where eq. (B.8b) is used for rewriting eq. (B.29b).

B.2.3.6. Kullback-Leibler Divergence

The relative entropy or Kullback-Leibler (KL) divergence between two distributions $p(\mathbf{x})$ and $q(\mathbf{x})$ is defined as

$$\mathrm{KL}(p\,||\,q) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})}\, d\mathbf{x} \tag{B.30}$$

which is always non-negative and zero only if $p = q$ (Cover and Thomas, 1991, ch. 2.3). Note the asymmetry in the definition. The Kullback-Leibler divergence between two Gaussian distributions $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ in nats is:

$$\mathrm{KL}(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\,||\,\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)) = \int \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \ln \frac{\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)}\, d\mathbf{x}$$

$$= (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}_2^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) - \tfrac{1}{2}\ln\left|\boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_2^{-1}\right| + \tfrac{1}{2}\,\mathrm{tr}(\boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_2^{-1} - \mathbf{I})\,. \tag{B.31}$$

B.2.4. Generating Samples of a Multivariate Normal Distribution

Generating samples from a multivariate normal distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be implemented if a source of univariate standard normal samples $\mathcal{N}(y|0, 1)$ is available, as provided by many common programming environments. Let $\mathbf{y}$ denote a sample of $\mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{I})$ then by using a linear transformation:

$$\mathbf{x} = \boldsymbol{\Sigma}^{1/2}\mathbf{y} + \boldsymbol{\mu} \tag{B.32}$$

we obtain a random sample distributed according to $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (see Appendix B.2.2.3). In case the $\boldsymbol{\Sigma}$ is full rank $\boldsymbol{\Sigma}^{1/2} = \mathbf{L}$ where $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top$ is a Cholesky decomposition (see Appendix B.1.5). For the singular multivariate normal distribution (3.22) $\boldsymbol{\Sigma}$ is only of rank $r < n$. In this case $\boldsymbol{\Sigma}^{1/2} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}$ where $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ is the eigendecomposition and $\boldsymbol{\Lambda}$ is an $[r \times r]$ matrix of nonzero eigenvalues. For details on generating random samples of standard distributions see for example Devroye (1986) or Hörmann et al. (2004).

C. Dynamical Systems

C.1. Mountain Car

The mountain car problem as described by Moore and Atkeson (1995) is to control a car in a frictionless landscape described by

$$H(x) \; = \; \begin{cases} x^2 + x & \text{for } x < 0 \\ \frac{x}{\sqrt{1+5x^2}} & \text{for } x \geq 0 \end{cases} \; . \tag{C.1}$$

The car has point mass $M = 1$. The state $\boldsymbol{s} = [x, \dot{x}]$ of the car is characterised by its position x and its velocity $\dot{x}$, which are constrained to $-1 \leq x \leq 1$ and $-2 \leq \dot{x} \leq 2$ respectively, see Figure 7.4 on page 141. The control action is a horizontal force F in the range $-4 \leq F \leq 4$. Let $H'(x) = \frac{d}{dx}H(x)$ and likewise $H''(x) = \frac{d^2}{dx^2}H(x)$, then Moore and Atkeson (1995) give the expression:

$$\ddot{x} \; = \; \frac{F}{M\sqrt{1 + (H'(x))^2}} - \frac{gH'(x)}{1 + (H'(x))^2} \tag{C.2}$$

to describe the dynamics where $g = 9.81$ is the gravity constant. In our experiments we use a corrected version thereof:

$$\ddot{x} \; = \; \frac{F}{M\sqrt{1 + (H'(x))^2}} - \frac{gH'(x)}{1 + (H'(x))^2} - \frac{\dot{x}^2 H'(x) H''(x)}{(1 + (H'(x))^2)} \tag{C.3}$$

which takes centripetal forces into account. In the experiments the force is applied constantly for a fixed time interval Δt.

C.2. Inverted Pendulum

The state of the system is a quadruple $\boldsymbol{s} = [x, \dot{x}, \theta, \dot{\theta}]$ of the cart's position x, its speed $\dot{x}$, the angle of the pendulum θ relative to pointing exactly downwards, and the angular speed $\dot{\theta}$, see Figure 7.8 on page 145. In the above experiments we parameterised the cart to have a mass of $M = 0.5$kg and the mass of the pendulum was set to be $m = 0.5$kg. The length of the pendulum is $l = 0.3$m and the inertia was set to $I = 0.06\text{kg} \cdot \text{m}^2$. The friction between the cart and the ground was set to 0.1N/m/sec while the joint has been assumed frictionless. The control signal is a horizontal force F which can be chosen every $\Delta t = 0.2$sec for which it is applied constantly. Given the two equations of motion

$$\begin{aligned}(M+m)\,\ddot{x}+b\,\dot{x}+m\,l\,\ddot{\theta}\cos\theta-m\,l\,\dot{\theta}^2\sin\theta &= F &\text{(C.4a)}\\ \left(I+m\,l^2\right)\ddot{\theta}+m\,g\,l\sin\theta &= -m\,l\,\ddot{x}\cos\theta &\text{(C.4b)}\end{aligned}$$

the dynamics were simulated by solving the system of differential equations

$$\begin{aligned}\ddot{x} &= \left(F-b\dot{x}+\frac{m^2l^2g\sin\theta\cos\theta}{\left(I+ml^2\right)}+ml\dot{\theta}^2\sin\theta\right)\left(\frac{\left(I+ml^2\right)}{(M+m)\left(I+ml^2\right)-m^2l^2\cos^2\theta}\right) &\text{(C.5a)}\\ \ddot{\theta} &= \left(ml\cos\theta\left[-F+b\dot{x}-ml\dot{\theta}^2\sin\theta\right]-mgl\sin\theta\right)\left(\frac{1}{(M+m)\left(I+ml^2\right)-m^2l^2\cos^2\theta}\right) &\text{(C.5b)}\end{aligned}$$

using the MATLAB implementation of the Runge-Kutta method.

Bibliography

P. Abrahamsen. A review of Gaussian random fields and correlation functions. Technical Report 917, Norwegian Computing Center, Oslo, 1997.

M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions.* Dover, New York, 1965.

R. J. Adler. *The Geometry of Random Fields.* John Wiley & Sons, Chichester, 1981.

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society, Series B*, 36(1):99–102, 1974.

A. Arapostathis, V. S. Borkar, E. Fernández-Gaucherand, M. K. Ghosh, and S. I. Marcus. Discrete-time controlled Markov processes with average cost criterion: A survey. *SIAM Journal of Control and Optimization*, 31(2):282–344, 1993.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

G. A. Barnard. Studies in the history of probability and statistics: IX. Thomas Bayes's essay towards solving a problem in the doctrine of chances. *Biometrika*, 45(3/4): 293–315, 1958.

M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference.* PhD thesis, Gatsby Computational Neuroscience Unit, University College London, UK, 2003.

S. Becker, S. Thrun, and K. Obermayer, editors. *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2003. The MIT Press.

R. E. Bellman. *Introduction to the Mathematical Theory of Control Processes*, volume 1. Academic Press, New York, 1967.

R. E. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, sixth edition, 1972.

J. O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer, New York, second edition, 1985.

J. O. Berger. An overview of robust Bayesian analysis. Technical Report #93-53C, Department of Statistics, Purdue University, 1991.

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory.* John Wiley & Sons, Chichester, 1994.

D. A. Berry and B. Fristedt. *Bandit Problems.* Chapman & Hall, New York, 1985.

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA, 1996.

C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1995.

C. M. Bishop, editor. *Neural Networks and Machine Learning*, volume 168 of *NATO ASI Series.* Springer, Berlin, 1998.

B. J. N. Blight and L. Ott. A Bayesian approach to model inadequacy for polynomial regression. *Biometrika*, 62(1):79–88, 1975.

G. E. P. Box. Robustness in the strategy of scientific model building. In R. L. Launer and G. N. Wilkinson, editors, *Robustness in Statistics*, pages 201–236. Academic Press, New York, 1979.

G. E. P. Box and G. C. Tiao. A further look at robustness via Bayes's theorem. *Biometrika*, 49(3/4):419–432, 1962.

G. E. P. Box and G. C. Tiao. A note on criterion robustness and inference robustness. *Biometrika*, 51(1/2):169–173, 1964.

G. E. P. Box and G. C. Tiao. A Bayesian approach to some outlier problems. *Biometrika*, 55(1):119–129, 1968.

G. E. P. Box and G. C. Tiao. *Bayesian Inference in Statistical Analysis.* Addison-Wesley, Reading, 1973.

S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, UK, 2004.

S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1–3):33–57, 1996.

T. Briegel and V. Tresp. Robust neural network regression for offline and online learning. In Solla et al. (2000).

C.-C. Chang and C.-J. Lin. *LIBSVM: A library for Support Vector Machines*, 2001. `http://www.csie.ntu.edu.tw/∼cjlin/libsvm`.

O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1):131–159, 2002.

W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.

W. Chu, S. S. Keerthi, and C. J. Ong. Bayesian support vector regression using a unified loss function. *IEEE Transactions on Neural Networks*, 15(1):29–44, 2004.

P. Cipollini, G. Corsini, M. Diani, and R. Grasso. Retrival of sea water optically active parameters from hyperspectral data by means of generalized radial basis function neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 39(7):1508–1524, 2001.

D. Collet. *Modelling Binary Data.* Chapman & Hall, Boca Raton, 1991.

J. B. Copas. Binary regression models for contaminated data. *Journal of the Royal Statistical Society, Series B*, 50(2):225–265, 1988.

T. M. Cover and J. A. Thomas. *Information Theory.* John Wiley & Sons, New York, 1991.

M. K. Cowles and B. P. Carlin. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904, 1996.

R. T. Cox. Probability, frequency, and reasonable expectation. *American Journal of Physics*, 14:1–13, 1946.

N. A. C. Cressie. *Statistics for Spatial Data.* John Wiley & Sons, New York, 1993.

L. Csató and M. Opper. Sparse online Gaussian processes. *Neural Computation*, 14(2): 641–669, 2002.

A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Applications.* Cambridge University Press, Cambridge, UK, 1997.

P. Dayan. The convergence of TD(λ) for general λ. *Machine Learning*, 8(3–4):341–362, 1992.

P. Dayan and T. J. Sejnowski. TD(λ) converges with probability 1. *Machine Learning*, 14(3):295–301, 1994.

R. Dearden, N. Friedman, and S. Russell. Bayesian $\mathcal{Q}$-learning. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 1998.

M. H. DeGroot and M. J. Schervish. *Probability and Statistics.* Addison-Wesley, Boston, third edition, 2002.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1): 1–38, 1977.

L. Devroye. *Non-Uniform Random Variate Generation.* Springer, New York, 1986.

H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support vector regression machines. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 155–161, Cambridge, MA, 1997. The MIT Press.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.

M. O. Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes.* PhD thesis, University of Massachusetts, Amherst, MA, 2002.

S. Efromovich. *Nonparametric Curve Estimation.* Springer, New York, 1999.

Y. Engel, S. Mannor, and R. Meir. Bayes meets Bellman: The Gaussian process approach to temporal difference learning. In T. Fawcett and N. Mishra, editors, *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003).* AAAI Press, 2003.

Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with Gaussian processes. In L. De Raedt anf S. Wrobel, editor, *Proceedings of the 22nd International Conference on Machine Learning*, pages 201–208, ACM Press, 2005.

A. C. Faul and M. E. Tipping. A variational approach to robust regression. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proceedings of the International Conference on Artificial Neural Networks*, pages 95–102. Springer, 2001.

C. Fernandez and M. F. J. Steel. Multivariate Student-t regression models: Pitfalls and inference. *Biometrika*, 86(1):153–167, 1999.

S. E. Fienberg. When did Bayesian inference become Bayesian? *Bayesian Analysis*, 1 (1):1–40, 2006.

J. Filar and K. Vrieze. *Competitive Markov Decision Processes.* Springer, New York, 1996.

R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A*, 222:309–368, 1922.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19 (1):1–67, 1991.

S. Geisser and W. F. Eddy. A predictive approach to model selection. *Journal of the American Statistical Association*, 74(365):153–160, 1979.

A. Gelman. Inference and monitoring convergence. In Gilks et al. (1996), pages 131–143.

A. Gelman and X.-L. Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13(2):163–185, 1998.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, Boca Raton, 1995.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

M. G. Genton. Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312, 2001.

J. Geweke. Bayesian treatment of the independent Student-t linear model. *Journal of Applied Econometrics*, 8:S19–S40, 1993. Supplement: Special Issue on Econometric Inference Using Simulation Techniques.

Z. Ghahramani and M. J. Beal. Graphical models and variational methods. In M. Opper and D. Saad, editors, *Advanced Mean Field Methods: Theory and Practice*, pages 161–178. The MIT Press, Cambridge, MA, 2000.

Z. Ghahramani and M. J. Beal. Propagation algorithms for variational Bayesian learning. In Leen et al. (2001), pages 507–513.

M. N. Gibbs and D. J. C. MacKay. Variational Gaussian process classifiers. *IEEE Transactions on Neural Networks*, 11(6):1458–1464, 2000.

M. N. Gibbs and D. J. C. MacKay. Efficient implementation of Gaussian processes. Technical report, Department of Physics, Cavendish Laboratory, Cambridge University, 1997.

W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, Boca Raton, 1996.

A. Girard, C. Rasmussen, J. Quiñonero-Candela, and R. Murray-Smith. Multiple-step ahead prediction for non linear dynamic systems - a Gaussian process treatment wih propagation of the uncertainty. In Becker et al. (2003).

M. Girolami and S. Rogers. Variational Bayesian multinomial probit regression with Gaussian process priors. Technical Report TR-2005-205, Department of Computing Science, University of Glasgow, 2005.

J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B*, 41(2):148–177, 1979.

J. C. Gittins. *Multi-armed Bandit Allocation Indices*. John Wiley & Sons, Chichester, 1989.

G. H. Golub and C. F. Van Loan. *Matrix Computations.* John Hopkins University Press, Baltimore, second edition, 1989.

G. H. Golub and J. H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of Computation*, 23(106):221–230, 1969.

I. J. Good. Significance tests in parallel and in series. *Journal of the American Statistical Association*, 53(284):799–813, 1958.

R. F. Green. Outlier-prone and outlier-resistant distributions. *Journal of the American Statistical Association*, 71(354):502–505, 1976.

L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression.* Springer, New York, 2002.

M. S. Handcock and M. L. Stein. A Bayesian analysis of kriging. *Technometrics*, 35(4): 403–410, 1993.

W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models.* Springer, Berlin, 2004.

D. Harrison and D. L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978.

D. A. Harville. *Matrix Algebra From a Statistican's Perspective.* Springer, New York, 1997.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, New York, 2001.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

R. G. Haylock and A. O'Hagan. On inference for outputs of computationally expensive algorithms with uncertainty on the inputs. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 5*, pages 629–637. Oxford University Press, Oxford, UK, 1996.

R. Herbrich. *Learning Kernel Classifiers.* The MIT Press, Cambridge, MA, 2002.

R. Herbrich, T. Graepel, and C. Campbell. Bayes point machines. *Journal of Machine Learning Research*, 1:245–279, 2001.

S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998. `http://www.ics.uci.edu/∼mlearn/MLRepository.html`.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

W. Hörmann, J. Leydold, and G. Derflinger. *Automatic Nonuniform Random Variate Generation*. Springer, Heidelberg, 2004.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

R. A. Howard. Information value theory. *IEEE Transactions on System Science and Cybernetics*, 2(1):22–26, 1966.

P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

P. J. Huber. *Robust Statistics*. John Wiley & Sons, New York, 1981.

J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

T. S. Jaakkola and M. I. Jordan. Bayesian parameter estimation via variational mehods. *Statistics and Computing*, 10:25–37, 2000.

E. T. Jaynes. *Probability Theory*. Cambridge University Press, Cambridge, UK, 2003.

E. T. Jaynes. Bayesian methods: General background. In J. H. Justice, editor, *Maximum-Entropy and Bayesian Methods in Applied Statistics*, pages 1–25. Cambridge University Press, Cambridge, UK, 1986.

H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A*, 186(1007):453–461, 1946.

W. H. Jeffreys and J. O. Berger. Ockham's razor and Bayesian analysis. *American Scientist*, 80:64–72, 1992.

M. I. Jordan, editor. *Learning in Graphical Models*. The MIT Press, Cambridge, MA, 1999.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In Jordan (1999), pages 105–161.

L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

R. E. Kass, B. P. Carlin, A. Gelman, and R. M. Neal. Marov chain Monte Carlo in practice: A roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

M. G. Kendall. *The Advanced Theory of Statistics*, volume I. Charles Griffin & Co., London, fifth edition, 1952.

H. Kesten and F. Spitzer. Controlled Markov chains. *The Annals of Probability*, 3(1): 32–40, 1975.

H.-C. Kim and Z. Ghahramani. The EM-EP algorithm for Gaussian process classification. In *Proceedings of the Workshop on Probabilistic Graphical Models for Classification (at ECML)*, 2003.

G. S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2):495–502, 1970.

I. Kononenko and I. Bratko. Information-based evaluation criterion for classifier's performance. *Machine Learning*, 6(1):67–80, 1991.

D. G. Krige. A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52(6):119–139, 1951.

M. Kuss. Nonlinear multivariate analysis with geodesic kernels. Master's thesis, Technische Universität Berlin, 2002.

M. Kuss and C. E. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.

M. Kuss and C. E. Rasmussen. Assessing approximations for Gaussian process classification. In Weiss et al. (2006).

M. Kuss, F. Jäkel, and F. A. Wichmann. Bayesian inference for psychometric functions. *Journal of Vision*, 5(5):478–492, 2005a.

M. Kuss, T. Pfingsten, L. Csató, and C. E. Rasmussen. Approximate inference for robust Gaussian process regression. Technical Report 136, Max Planck Institute for Biological Cybernetics, Tübingen, 2005b.

K. L. Lange, R. J. A. Little, and J. M. G. Taylor. Robust statistical modelling using the t-distribution. *Journal of the American Statistical Association*, 84(408):881–896, 1989.

G. M. Laslett. Kriging and splines: An empirical comparison of their predictive performance in some applications. *Journal of the American Statistical Association*, 89 (426):391–409, 1994.

S. L. Lauritzen. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87(420):1098–1108, 1992.

N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In Becker et al. (2003), pages 609–616.

N. D. Lawrence and M. I. Jordan. Semi-supervised learning via Gaussian processes. In Saul et al. (2005), pages 753–760.

T. K. Leen, T. G. Dietterich, and V. Tresp, editors. *Advances in Neural Information Processing Systems 13*, Cambridge, MA, 2001. The MIT Press.

J. S. Liu. *Monte Carlo Strategies in Scientific Computing.* Springer, New York, 2001.

H. Lütkepohl. *Handbook of Matrices.* John Wiley & Sons, Chichester, 1996.

D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, Cambridge, UK, 2003.

D. J. C. MacKay. *Bayesian Methods for Adaptive Models.* PhD thesis, California Institute of Technology, Pasadena, 1992a.

D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992b.

D. J. C. MacKay. Introduction to Gaussian processes. In Bishop (1998), pages 133–165.

D. J. C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Compuration*, 11(5):1035–1068, 1999a.

D. J. C. MacKay. Introduction to Monte Carlo methods. In Jordan (1999), pages 175–204.

J. R. Magnus and H. Neudecker. *Matrix Differential Calculus.* John Wiley & Sons, Chichester, 1988.

K. V. Mardia and R. J. Marshall. Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, 71(1):135–146, 1984.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis.* Academic Press, London, 1979.

J. J. Martin. *Bayesian Decision Problems and Markov Chains.* John Wiley & Sons, New York, 1967.

D. A. McAllester. PAC-Bayesian model averaging. In *Proceedings of the 12th Annual Conference on Computational Learning Theory*, pages 164–170, New York, 1999. ACM Press.

P. McCullagh and J. A. Nelder. *Generalized Linear Models.* Chapman & Hall, Boca Raton, second edition, 1989.

T. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London, Series A*, 209:415–446, 1909.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Proceedings of the 1999 IEEE Workshop on Neural Networks for Signal Processing*, pages 41–48, Piscataway, NJ, 1999. IEEE.

T. P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, 2001a.

T. P. Minka. Expectation propagation for approximate Bayesian inference. In J. S. Breese and D. Koller, editors, *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann, 2001b.

M. L. Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

A. W. Moore and C. G. Atkeson. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. *Machine Learning*, 21(3): 199–233, 1995.

S. C. Narula and J. F. Wellington. The minimum sum of absolute errors regression: A state of the art survey. *International Statistical Review*, 50(3):317–326, 1982.

R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, New York, 1996.

R. M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report 9702, Department of Statistics, University of Toronto, 1997.

R. M. Neal. Regression and classification using Gaussian process priors. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 6*, pages 475–501. Oxford University Press, 1998a.

R. M. Neal. Assessing relevance determination methods using DELVE. In Bishop (1998), pages 97–129.

A. O'Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society, Series B*, 40(1):1–42, 1978.

A. O'Hagan. On outlier rejection phenomena in Bayes inference. *Journal of the Royal Statistical Society, Series B*, 41(3):358–367, 1979.

A. O'Hagan. *Bayesian Inference*, volume 2B of *Kendall's Advanced Theory of Statistics*. Arnold, London, 1994.

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

D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49 (2):161–178, 2002.

C. J. Paciorek and M. J. Schervish. Nonstationary covariance functions for Gaussian process regression. In Thrun et al. (2004).

R. Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955.

J. C. Platt. Probabilities for SV machines. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–73. The MIT Press, Cambridge, MA, 2000.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipies in C++*. Cambridge University Press, Cambridge, UK, second edition, 2002.

Y. Qi, T. P. Minka, R. W. Picard, and Z. Ghahramani. Predictive automatic relevance determination by expectation propagation. In R. Greiner and D. Schuurmans, editors, *Proceedings of Twenty-first International Conference on Machine Learning*, pages 671–678, 2004.

J. Quiñonero-Candela, A. Girard, J. Larsen, and C. E. Rasmussen. Propagation of uncertainty in Bayesian kernel models - application to multiple-step ahead forecasting. In *Proceedings of the 2003 IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP 03)*, 2003.

A. E. Raftery and S. M. Lewis. Implementing MCMC. In Gilks et al. (1996), pages 115–130.

C. E. Rasmussen. *Evaluation of Gaussian Processes and other Methods for Non-linear Regression*. PhD thesis, Department of Computer Science, University of Toronto, 1996.

C. E. Rasmussen and Z. Ghahramani. Occam's razor. In Leen et al. (2001), pages 294–300.

C. E. Rasmussen and M. Kuss. Gaussian processes in reinforcement learning. In Thrun et al. (2004).

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, Cambridge, MA, 2006.

R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

B. D. Ripley. *Pattern Recognition and Neural Newtorks.* Cambridge University Press, Cambridge, UK, 1996.

B. D. Ripley. Statistical theories of model fitting. In Bishop (1998), pages 3–25.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

C. P. Robert. *The Bayesian Choice.* Springer, New York, 1994.

P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection.* John Wiley & Sons, 1987.

J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

S. Saitoh. *Theory of Reproducing Kernels and its Applications.* Pitman Research Notes in Mathematics. Longman, Essex, 1988.

A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research*, 3:211–229, 1959.

L. K. Saul, Y. Weiss, and L. Bottou, editors. *Advances in Neural Information Processing Systems 17*, Cambridge, MA, 2005. The MIT Press.

G. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the 15th International Conference on Machine Learning*, pages 515–521, San Francisco, 1998. Morgan Kaufmann.

M. J. Schervish. *Theory of Statistics.* Springer, New York, 1997.

A. M. Schmidt and A. O'Hagan. Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3):745–758, 2003.

I. J. Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44(3):522–536, 1938.

B. Schölkopf and A. J. Smola. *Learning with Kernels.* The MIT Press, Cambridge, MA, 2002.

M. Seeger. Bayesian model selection for support vector machines, Gaussian processes and other kernel classifiers. In Solla et al. (2000), pages 603–609.

M. Seeger. PAC-Bayesian generalisation error bounds for Gaussian process classification. *Jounral of Machine Learning Research*, 3:233–269, 2002.

M. Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations.* PhD thesis, Institute of Adaptive and Neural Computation, University of Edinburgh, 2003.

M. Seeger. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14(2):69–106, 2004.

M. Seeger and M. I. Jordan. Sparse Gaussian process classification with multiple classes. Technical Report TR 661, Department of Statistics, University of California at Berkeley, 2004.

V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3):262–266, 1989.

E. Snelson, C. E. Rasmussen, and Z. Ghahramani. Warped Gaussian processes. In Thrun et al. (2004).

S. A. Solla, T. K. Leen, and K.-R. Müller, editors. *Advances in Neural Information Processing Systems 12*, Cambridge, MA, 2000. The MIT Press.

P. Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine Learning*, 46(1–3):21–52, 2002.

M. L. Stein. *Interpolation of Spatial Data.* Springer, New York, 1999.

S. M. Stigler. Laplace's 1774 memoir on inverse probability. *Statistical Science*, 1(3): 350–363, 1986.

S. M. Stigler. *Statistics on the Table: The History of Statistical Concepts and Methods.* Harvard University Press, Cambridge, MA, 1999.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36(2):111–147, 1974.

M. Strens. A Bayesian framework for reinforcement learning. In P. Langley, editor, *Proceedings of the seventeenth international conference on machine learning (ICML-2000)*, pages 943–950, San Francisco, CA, 2000. Morgan Kaufmann.

S. Sundararajan and S. S. Keerthi. Predictive approaches for choosing hyperparameters in Gaussian processes. *Neural Computation*, 13:1103–1118, 2001.

R. S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning*, pages 216–224. Morgan Kaufmann, 1990.

R. S. Sutton. Integrated modeling and control based on reinforcement learning and dynamic programming. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 3*, pages 471–478, 1991.

R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

R. S. Sutton and A. G. Barto. *Reinforcement Learning*. The MIT Press, Cambridge, MA, 1998.

J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

J. A. K. Suykens, J. Vandewalle, and B. De Moor. Optimal control by least squares support vector machines. *Neural Networks*, 14:23–35, 2001.

G. Tesauro. Programming backgammon using self-teaching neural nets. *Artificial Intelligence*, 134(1–2):181–199, 2002.

S. Thrun, L. Saul, and B. Schölkopf, editors. *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. The MIT Press.

M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

M. E. Tipping and N. D. Lawrence. A variational approach to robust Bayesian interpolation. In C. Molina, T. Adali, J. Larsen, M. V. Hulle, S. C. Douglas, and J. Rouat, editors, *Proceedings of the IEEE 2003 Neural Networks for Signal Processing Workshop*, pages 229–238, Piscataway, New Jersey, 2003. IEEE Press.

M. E. Tipping and N. D. Lawrence. Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis. *Neurocomputing*, 69(1–3): 123–141, 2005.

D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors. *Advances in Neural Information Processing Systems 8*, Cambridge, MA, 1996. The MIT Press.

J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

J. W. Tukey. Unsolved problems of experimental statistics. *Journal of the American Statistical Association*, 49(268):706–731, 1954.

J. W. Tukey. The future of data analysis. *The Annals of Mathematical Statistics*, 33 (1):1–67, 1962.

J. W. Tukey. The technical tools of statistics. *The American Statistician*, 19(2):23–28, 1965.

V. N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, New York, 1998.

V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, second edition, 1999.

G. Wahba. *Spline Models for Observational Data.* Society for Industrial and Applied Mathematics, Philadelphia, 1990.

M. D. Waltz and K. S. Fu. A heuristic approach to reinforcement learning control systems. *IEEE Transactions on Automatic Control*, 10(4):390–398, 1965.

C. J. C. H. Watkins. *Learning from Delayed Rewards.* PhD thesis, King's College, Cambridge, UK, 1989.

C. J. C. H. Watkins and P. Dayan. $\mathcal{Q}$-Learning. *Machine Learning*, 8(3–4):279–292, 1992.

Y. Weiss, B. Schölkopf, and J. Platt, editors. *Advances in Neural Information Processing Systems 18*, Cambridge, MA, 2006. The MIT Press.

M. West. Outlier models and prior distributions in Bayesian linear regression. *Journal of the Royal Statistical Society, Series B*, 46(3):431–439, 1984.

M. West. On scale mixtures of normal distributions. *Biometrika*, 74(3):646–648, 1987.

D. J. White. *Markov Decision Processes.* John Wiley & Sons, Chichester, 1993.

C. K. I. Williams. Computation with infinite neural networks. *Neural Computation*, 10: 1203–1216, 1998.

C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In Jordan (1999), pages 599–621.

C. K. I. Williams and D. Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In Touretzky et al. (1996), pages 514–520.

S. Wood and R. Kohn. A Bayesian approach to robust binary nonparametric regression. *Journal of the American Statistical Association*, 93(441):203–213, 1998.

A. M. Yaglom. *Stationary Random Functions.* Prentice-Hall, Englewood Cliffs, NJ., 1962.