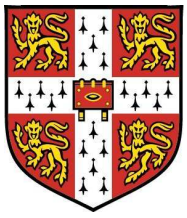
Sparse and robust kernel methods



Andrew Naish-Guzman

Gonville & Caius College

University of Cambridge

a thesis submitted for the degree of Doctor of Philosophy

December 2007

Declaration

I hereby declare that my thesis entitled *Sparse and robust kernel methods* is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. It is not substantially the same as any that I have submitted for a degree or diploma or other qualification at any other University, and does not exceed 60,000 words.

Andrew G. P. Naish-Guzman

Abstract

Statistical inference inescapably involves dealing with uncertainty, whether "out there" in the real world, or in the parameters of a hypothetical model. A powerful framework built upon the foundation of Bayes' theorem provides a set of mathematically sound rules for manipulating a probabilistic representation of belief in the light of new evidence. The Gaussian process (GP) is a widely extensible class of models to which all the mechanics of Bayesian inference may be applied: uncertainties are propagated into predictions for "error bars", and the evidence framework can be employed as a principled method for learning parameters higher in the model hierarchy.

This thesis explores methods that allow the wider application of GPs by addressing issues of complexity and robustness. Computationally, the GP becomes unwieldy for modelling relationships between more than a few hundred points due to the cubic scaling of inference and quadratic scaling of prediction times. A generic algorithm is presented, applicable to any sampling distribution, that reduces these costs to $\mathcal{O}(NM^2)$ and $\mathcal{O}(M^2)$ for N data points, using a small "active set" of size M. Supervised dimensionality reduction is investigated as a possible source of further advantage, and the model is evaluated for the common task of binary classification.

For the convenience of a fully tractable GP inference problem, it is often assumed that data have been corrupted with independent and identically distributed Gaussian noise. In domains of heavy-tailed or heteroscedastic corruptions this assumption is inappropriate. A new noise model is presented in which fluctuations in sample variance are achieved by using a second GP to partition softly between two noise regimes. This design permits an efficient deterministic inference, avoiding the need for slow stochastic sampling of the posterior. The faithfulness of the result using an approximate posterior is compared to that returned by Monte Carlo integration. The model is further shown to come from a certain class of GP mixtures; applications of the design to classification and more general regression and mixture modelling tasks are discussed, in each of which the benefit of deterministic inference is retained.

Acknowledgments

I owe several people my gratitude for their assistance in my research and their contributions to my development as a postgraduate student. I am particularly indebted to Ed Snelson, not only for his novel Gaussian process model which forms the heart of the second chapter of this dissertation, but also for some illuminating discussions, and even a few LaTeX hints in the small hours before a submission deadline.

Carl Rasmussen was instrumental in helping me refine my approach to generalizing the FITC approximation, for which I am very grateful. I would also like to express my sincere thanks to Neil Lawrence for his generous provision of feedback on queries regarding the IVM, and for suggestions regarding a conference submission. I acknowledge too Malte Kuss for his prompt and very helpful responses to some technical questions about his work on robust Gaussian process regression, and Oli Stegle for discussions which originally inspired my interest in that area.

Within the Computer Laboratory, I thank Sean Holden, who allowed me considerable freedom to explore areas of personal interest, and Ulrich Paquet, a friendly face in the office for three years and a great source of insight and humour. In recognition of the work involved in organising the various machine learning lectures, reading groups and events that I attended during my time here, I would like to thank David MacKay, Zoubin Ghahramani and Martin Szummer.

Beyond the academic world, I thank my friends at Caius for filling these graduate years with so many very happy memories, and yet further afield, three women without whom I might still be writing: Fiona, a miraculous package of affection and encouragement, Ileana, a loving and attentive sister, and my mother, whose patience and support were as invaluable as they were inexhaustible.

Andrew G. P. Naish-Guzman
Cambridge, December 2007

Contents

List of Figures

List of Tables

List of Acronyms

BCM	Bayesian committee machine
EM	Expectation maximization
EP	Expectation propagation
FI(T)C	Fully independent (training) conditional
GP(C)	Gaussian process (classifier)
IVM	Informative vector machine
MCMC	Markov chain Monte Carlo
ML	Maximum likelihood
PI(T)C	Partially independent (training) conditional
PP *or* PLV	Projected process *or* projected latent variables
RBF	Radial basis function
SD	Subset of data
SPGP	Sparse pseudo-input Gaussian process
SVM	Support vector machine
TGP	Twinned Gaussian process
VB	Variational Bayes

CHAPTER 1

Introduction

As human beings, at every waking moment we are bombarded with raw data by our senses. From our very earliest days we recognise the familiar faces around us; from great distances we distinguish a smile from a grimace; and from its opening chords we identify a favourite symphony. Not only do our brains extract salient features and process them with an extraordinary level of accuracy, they do so remarkably quickly and with a profound tolerance of artefacts and other noise. For millennia, these exceptional faculties have been the exclusive preserve of higher intelligent life, yet now in restricted domains we can reproduce some of this remarkable processing without recourse to organic grey matter.

Recent developments and parallels in fields as disparate as information theory, statistical physics, computational learning theory, graph theory and computer science have yielded a wealth of theoretical analyses and a library of algorithms for mechanical inference. These can be brought to bear on datasets which humans struggle to classify or visualize, and to augment the power of human deduction with the reliability of a machine's calculations: many credit card applications are seen only by an artificial neural network; handwritten digit recognition is commonplace in banks; sentient computing is entering the mainstream, with human-computer interaction assisted by

emotion recognition. Evidently, the range of applications of machine learning is almost limitless, but this presents the theoretician with a challenge: how can one avoid ad hoc, domain-dependent solutions, and achieve a unified approach to learning?

This thesis is about a class of probabilistic models called Gaussian processes. Although employed in stochastic modelling since the 1940s, and later used in the field of geostatistics under the name *kriging*, their wider relevance was not realised until the mid-1990s, since when they have enjoyed a surge in popularity amongst researchers and practitioners alike. The Gaussian process provides one solution to the problem of domain-dependent algorithms: prior knowledge may readily be incorporated in the form of the *kernel*—a measure of correlation that lies at the core of all such models—but the underlying probabilistic machinery and methodology remain universally consistent. The price of this flexibility is paid in the complexity of the inference; an additional cost is the non-robustness of the standard model to large errors. We hope to extend the class of Gaussian processes without sacrificing the versatility of the original design by addressing each of these issues; in more ambitious terms, the work is an attempt to make the framework more akin to the efficient, resilient and endlessly adaptable inference system carried by each of us.

We begin with an introduction to Bayesian probability, the foundation of all probabilistic inference in machine learning, before considering the Gaussian process in detail. Subsequent sections present some common methods for approximate Bayesian inference, and we conclude the chapter with an overview of the structure of the thesis.

1.1 The Bayesian paradigm

The classical theory of probability assigns to real numbers between 0 and 1 a frequentist interpretation based on the limit of repeated samples. These semantics restrict the scope of applicability to repeatable events: "there's a 50% chance it will rain tomorrow" is a meaningless statement to the frequentist since tomorrow can happen only once. The Bayesian is more flexible: he interprets such numbers as subjective *degrees of belief* that are manipulated consistently to derive new beliefs from evidence; they can be applied sensibly to statements about the weather, or indeed to any other event or object—or model parameter, as discussed below—about which we hold beliefs. Formal justification for a Bayesian perspective is given by Cox (1946), from whose postulates about reasonable belief can be derived a logical interpretation

of probability mathematically equivalent to the frequentist or measure-theoretic view (see also Jaynes, 2003, ch. 2). The subjective nature of Bayesian reasoning is the foundation which many frequentists attack as arbitrary. It is also a strength: the Bayesian paradigm requires its practitioners to make explicit all their assumptions in the form of prior beliefs, assumptions which are typically less apparent in the statistical tests of a frequentist. It also permits for the computer scientist a powerful and very general approach to machine learning: the refinement of beliefs in the presence of data.

1.1.1 Bayes' theorem

Imagine we have a probabilistic model $\mathcal{M}$ that describes a random process of data generation, with a set of parameters $\boldsymbol{\psi}$—for example, $\mathcal{M}$ is a Gaussian distribution and $\boldsymbol{\psi}$ its mean and variance. We might ask what information we learn about $\boldsymbol{\psi}$ from observations D. This question is usually interpreted by the frequentist as "what are the most likely parameters to have generated the data?", but Bayes' theorem answers the question directly via the simple yet profound rearrangement of an identity from probability theory (we drop the implicit dependence on $\mathcal{M}$):

$$p(\boldsymbol{\psi}|D) = \frac{p(D|\boldsymbol{\psi})p(\boldsymbol{\psi})}{p(D)}. \tag{1.1}$$

The term $p(\boldsymbol{\psi})$ is the *prior*, our beliefs about the model parameters before we have observed any data. On the left-hand side is the *posterior*, our beliefs updated consistently with our knowledge of the model. The term $p(D|\boldsymbol{\psi})$ is the sampling distribution, a function of $\boldsymbol{\psi}$ called the *likelihood*; it defines the probability of generating the observed data D using model $\mathcal{M}$ with parameters $\boldsymbol{\psi}$.

Having revised our beliefs, a common task is to make predictions about some unknown feature $f_\star$. Such inferences need to account for our incomplete knowledge of the true generative parameter $\boldsymbol{\psi}$. The frequentist, unable to accommodate beliefs over model parameters, is restricted to a single $\hat{\boldsymbol{\psi}}$, usually chosen to minimize a particular loss function. Different choices of loss function will yield different optimal $\hat{\boldsymbol{\psi}}$: the expected squared error is minimized by the mean of the posterior; the expected absolute error by the median. The mode of the posterior is known as the *maximum a posteriori* (MAP) solution, and is optimal for a loss function that is zero at the true $\boldsymbol{\psi}$ and a positive constant otherwise (Jaynes, 2003). It is also known as penalized maximum likelihood by frequentists, since it can be viewed as an ad hoc refinement of the maximum likelihood method to discourage highly tuned parameters. The Bayesian has reservations

about the procedure: with an appropriate change of basis, any solution with non-zero prior probability can be obtained as the MAP solution. Furthermore, its focus is on the mode of the posterior, where the probability *density* can be very large but which is often highly unrepresentative of the distribution of posterior *mass*. Rather, the Bayesian would make use of the full posterior by *marginalizing* $\boldsymbol{\psi}$,

$$p(f_\star|\mathcal{M}) = \int p(f_\star|\boldsymbol{\psi})p(\boldsymbol{\psi}|D)\mathrm{d}\boldsymbol{\psi},$$

thereby incorporating into the prediction all the information provided by D.

1.1.2 The evidence

The denominator $p(D)$ in (1.1) normalizes the posterior distribution. It is known as the *evidence*; it is also called the *marginal likelihood* because we marginalize over the parameters $\boldsymbol{\psi}$ in its evaluation:

$$p(D|\boldsymbol{\theta}) = \int p(D|\boldsymbol{\psi})p(\boldsymbol{\psi}|\boldsymbol{\theta})\mathrm{d}\boldsymbol{\psi}. \tag{1.2}$$

We have explicitly introduced $\boldsymbol{\theta}$ here to denote optional *hyperparameters* that control the distribution of the prior: Bayes' theorem can now be applied hierarchically by treating (1.2) as a likelihood, allowing us to refine beliefs about these hyperparameters after observing data; indeed, the unifying paradigm of Bayesian inference treats the model itself as just another parameter about which we maintain belief. If we compare two models under an equal prior belief, the ratio of the posterior odds is equal to the ratio of the evidences, or *Bayes factor* (Kass and Raftery, 1995)

$$\frac{p(\mathcal{M}_1|D)}{p(\mathcal{M}_2|D)} = \frac{p(D|\mathcal{M}_1)p(\mathcal{M}_1)}{p(D|\mathcal{M}_2)p(\mathcal{M}_2)}.$$

The evidence is thus of fundamental importance in model selection since it can be used as a proxy for the posterior belief. By interpreting it as a measure of generative ability, we see that overly rigid models are penalized if they cannot place significant probability mass at D for any parameterization, whereas very flexible models (by definition with a normalizing mass over a broad subset of the data space) cannot give significant weight to all possible data (see fig. 1.1). Model selection by evidence maximization therefore incorporates a natural *Occam's razor* effect, pruning complexity to leave a model sufficiently intricate to explain the observations, but without unnecessary embellishments.

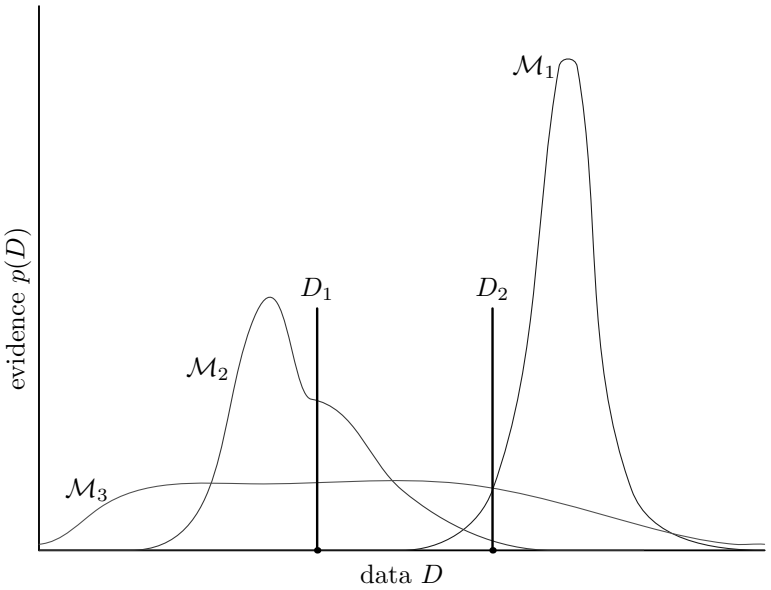


Figure 1.1: The evidence and Bayesian model selection. In this stylized example, if we observe data D_1, the model $\mathcal{M}_2$ is strongly favoured over the straitened $\mathcal{M}_1$, and by about 2:1 over the unnecessarily complex $\mathcal{M}_3$, which spreads its generative mass over a wide area of the data domain. However, were we to observe D_2 in the absence of prior information, the posterior belief in $\mathcal{M}_1$ and $\mathcal{M}_3$ would be approximately equal, with $\mathcal{M}_2$ rather implausible. Note that these relative considerations give no indication as to the absolute truth of a model.

Having chosen a single model, which can be considered a maximum likelihood (ML) procedure, it is also common to apply the ML principle at the level of hyperparameters $\boldsymbol{\theta}$ (known as "type-II ML") to determine their most plausible values. This procedure is much less prone to the overfitting that plagues ML techniques at lower levels $\boldsymbol{\psi}$ (MacKay, 1999) and avoids the further, often intractable integration demanded by the inflexible Bayesian. Although strictly type-II ML violates the Bayesian principle—we are adjusting the prior (as a function of $\boldsymbol{\theta}$) *after* observing data—its empirical success and computational advantages have made its use widespread in the community.

1.2 Gaussian processes

The Gaussian process (GP) is a popular and versatile tool for data interpolation. It has the advantage of a rigorous Bayesian foundation, providing full predictive distributions as opposed to point estimates and allowing a principled approach to model selection. Rather than define a non-linear function in terms of a set of weights with an associated prior (MacKay, 1991, 1992a), the GP framework places a prior directly on the space of functions, specifying the kinds of values we expect to observe in terms of their

mutual covariance. Fortunately we do not need to deal with all possible inputs: the Gaussian distribution enjoys a marginalization property that allows us effectively to ignore points not specified in the training and test sets and retain a Gaussian form.

In this section, the input domain is denoted $\mathcal{X}$ and the latent outputs are $\mathbf{f} = \{f_n\}_{n=1}^N$, where $f_n \in \mathbb{R}$. The training inputs are denoted $\mathbf{X}$ and test inputs $\mathbf{X}_\star$, with $\mathbf{X}, \mathbf{X}_\star \subset \mathcal{X}$. The targets in the regression case are real, for which we write $\mathbf{y} = \{y_n\}_{n=1}^N$, $y_n \in \mathbb{R}$; targets in the classification case are binary, $y_n \in \{\pm 1\}$.

1.2.1 The covariance function

GPs may be understood as the generalization of a Gaussian distribution to an infinite index set $\mathcal{X}$: formally, a GP is a collection of random variables indexed by elements of $\mathcal{X}$, any finite number of which have a jointly Gaussian distribution. To specify a GP, we need to define its mean (as a function of some input $\mathbf{x}$) and covariance (as a function of arbitrary $\mathbf{x}$ and $\mathbf{z}$). In most cases we will only be interested in zero-mean GPs, since constants and linear trends can be subtracted from the data in a preprocessing stage. The crucial component is then the covariance, which encapsulates all our beliefs about the function we wish to learn (its amplitude, lengthscales, smoothness, etc.). It is defined by a covariance or *kernel* function

$$k(\mathbf{x}, \mathbf{z}) \in \mathcal{X} \times \mathcal{X} \to \mathbb{R},$$

such that elements of a *covariance matrix* $K_{nn'} = \mathbb{E}[f(\mathbf{x}_n) f(\mathbf{x}_{n'})]$ are given by $k(\mathbf{x}_n, \mathbf{x}_{n'})$. The kernel describes the correlation between the random variables associated with the two indices, and more generally defines a notion of "closeness" in $\mathcal{X}$, since we expect the values of the latent function to be more correlated for closer points than distant points.

There is a restriction that any valid covariance function must be positive semi-definite. This guarantees that our distance measure corresponds to a dot product in some high dimensional feature space: intuitively, that it cannot provide inconsistent measurements; in particular, that the evaluation of the covariance function for all pairs $(\mathbf{x}, \mathbf{z})$ in $\mathbf{X}$ yields a covariance matrix $\mathbf{K}$ which is positive semi-definite, i.e. for which

$$\mathbf{v}^T \mathbf{K} \mathbf{v} \geq 0 \qquad \text{for all } \mathbf{v} \in \mathbb{R}^N.$$

The squared exponential

The most commonly used covariance function is the isotropic squared exponential

$$k_{\text{SE}}(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = a \exp\left\{-\frac{r^2}{2l^2}\right\}, \qquad \boldsymbol{\theta} = \{a, l\}, \tag{1.3}$$

where $r = \|\mathbf{x} - \mathbf{z}\|$, and $\boldsymbol{\theta}$ is a set of kernel parameters:[1] l defines the characteristic lengthscale of the process, while a controls the signal variance. We describe it as a *stationary* covariance function since its value depends only on the distance r between its inputs. Samples from a process with this prior are shown in fig. 1.2a with hyperparameters $a = 1$ and $l = 1$. An *anisotropic* covariance function may be preferable when the domain is multi-dimensional, especially if we do not believe all the lengthscales of the generative process are equal: the natural extension of the squared exponential to D dimensions is

$$k_{\text{aSE}}(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = a \exp\left\{-\sum_{d=1}^{D} \frac{(x_d - z_d)^2}{2l_d^2}\right\}, \qquad \boldsymbol{\theta} = \{a, l_1, l_2, \ldots, l_D\}. \tag{1.4}$$

The Matérn class

The squared exponential kernel gives rise to functions which are infinitely differentiable. Such smoothness is not always desired, and the Matérn class of covariances provides rougher samples. Their general form is

$$k_{\text{Mat}}(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{l}\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu} r}{l}\right), \qquad \boldsymbol{\theta} = \{\nu > 0, l > 0\},$$

where K_ν is a modified Bessel function (Abramowitz and Stegun, 1964, sec. 9.6). As $\nu \to \infty$ we recover the squared exponential; common values are $\nu = \frac{3}{2}$ and $\nu = \frac{5}{2}$ since half-integer values allow simplification to the definition, and larger values give processes increasingly indistinguishable from those of the squared exponential kernel. The special case of $\nu = \frac{1}{2}$ reduces to the covariance of the Ornstein-Uhlenbeck process used as a model for Brownian motion (Uhlenbeck and Ornstein, 1930).

[1] The kernel parameters should be identified with the hyperparameters of section 1.1; "parameters" in a GP, earlier denoted ψ, are the latent $\mathbf{f}$ that are everywhere marginalized.

Non-stationary covariance functions

Not all kernels must be functions only of the separation of their vector arguments. For example, the dot product

$$k_{\text{dot}}(\mathbf{x}, \mathbf{z}) = \mathbf{x} \cdot \mathbf{z}$$

is a valid and non-stationary kernel function. While typically inappropriate for regression, similar polynomial kernels have enjoyed success in classification problems (Burges and Schölkopf, 1997), where the input data are normalized to prevent the unbounded growth of the kernel output.

An early work which sparked interest in Gaussian processes amongst the machine learning community is Neal (1996). It shows that neural networks—then a very popular tool for semi-parametric modelling—in the limit of an infinite number of hidden units become GP models; Williams (1998) gave the first neural network covariance function, corresponding to an erf transfer function:

$$k_{\text{NN}}(\mathbf{x}, \mathbf{z}|\boldsymbol{\Sigma}) = \frac{2}{\pi} \arcsin\left(\frac{2\hat{\mathbf{x}}^T \boldsymbol{\Sigma} \hat{\mathbf{z}}}{\sqrt{(1 + 2\hat{\mathbf{x}}^T \boldsymbol{\Sigma} \hat{\mathbf{x}})(1 + 2\hat{\mathbf{z}}^T \boldsymbol{\Sigma} \hat{\mathbf{z}})}}\right).$$

The input $\hat{\mathbf{x}} = [1\, \mathbf{x}^T]^T$ is augmented to accommodate a bias, and $\boldsymbol{\Sigma}$ is a weight prior, typically $\mathsf{diag}\,(\sigma_0^2, \sigma_1^2, \ldots, \sigma_D^2)$. The first parameter controls the variance of the neural network bias, while the remaining parameters affect the scaling in each of the D dimensions.

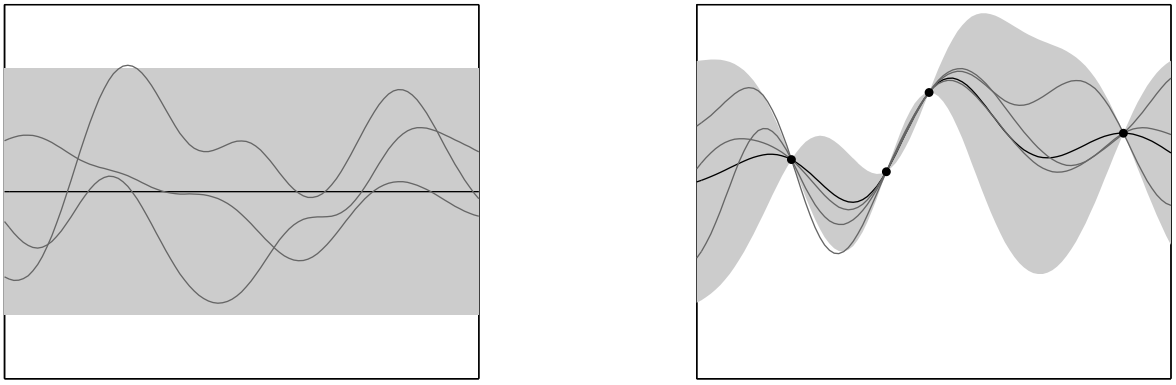
Individual kernels may be combined in various ways including addition, multiplication and convolution, to produce legitimate composite kernels. Other examples of valid covariance functions appear in Rasmussen and Williams (2006, sec. 4.2).

1.2.2 Gaussian process regression

GIven a zero-mean GP prior, the joint distribution of the latent function at training and test points is

$$p\left(\left[\begin{array}{c} \mathbf{f} \\ \mathbf{f}_\star \end{array}\right] \middle| \mathbf{X}, \mathbf{X}_\star\right) = \mathcal{N}\left(\left[\begin{array}{c} \mathbf{f} \\ \mathbf{f}_\star \end{array}\right]; \mathbf{0}, \left[\begin{array}{cc} \mathbf{K}_{\mathbf{ff}} & \mathbf{K}_{\mathbf{f}\star} \\ \mathbf{K}_{\star\mathbf{f}} & \mathbf{K}_{\star\star} \end{array}\right]\right),$$

where $\mathbf{K}_{\mathbf{ff}}$ ($\mathbf{K}_{\star\star}$) is the matrix of kernel evaluations between all pairs of training (test) data, and $\mathbf{K}_{\mathbf{f}\star}$ is the $N \times M$ matrix of evaluations $\{k(\mathbf{x}^{(n)}, \mathbf{x}_\star^{(m)})\}_{n=1,m=1}^{N,M}$; we switch the



(a) (b)

Figure 1.2: The covariance structure imposed by the GP prior in panel (a) yields samples (grey lines) with a characteristic lengthscale and magnitude. After observing data (black dots) in panel (b), the posterior GP is pulled away from zero: its mean (black line) passes near the observations, and its variance reduces in the vicinity of the data. The shaded area shows 95% confidence intervals.

indices to $\mathbf{K}_{\star\mathbf{f}}$ to indicate the transpose. In standard GP regression, we use a model in which observations y_n are i.i.d. corruptions of latent f_n by zero-mean Gaussian noise of variance σ^2: the joint distribution of $(\mathbf{y}, \mathbf{f}_\star)$ is

$$p\left(\left[\begin{array}{c}\mathbf{y}\\ \mathbf{f}_\star\end{array}\right]\middle|\mathbf{X}, \mathbf{X}_\star\right) = \mathcal{N}\left(\left[\begin{array}{c}\mathbf{y}\\ \mathbf{f}_\star\end{array}\right]; \mathbf{0}, \left[\begin{array}{cc}\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I} & \mathbf{K}_{\mathbf{f}\star}\\ \mathbf{K}_{\star\mathbf{f}} & \mathbf{K}_{\star\star}\end{array}\right]\right).$$

Conditioning on the observations at the training inputs we obtain a Gaussian posterior,[2] for which the distribution over the latent function values at the test inputs is

$$p(\mathbf{f}_\star|\mathbf{X}_\star, \mathbf{X}, \mathbf{y}) = \mathcal{N}\big(\mathbf{f}_\star\,;\, \mathbf{K}_{\star\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1}\mathbf{y}\,,\, \mathbf{K}_{\star\star} - \mathbf{K}_{\star\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{f}\star}\big)\,.$$

We recognise this as another GP, with a posterior non-zero mean function

$$m(\mathbf{x}_\star) = \mathbf{k}(\mathbf{X}, \mathbf{x}_\star)^T(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1}\mathbf{y},$$

and a covariance function

$$k(\mathbf{x}_\star, \mathbf{z}_\star) = k(\mathbf{x}_\star, \mathbf{z}_\star) - \mathbf{k}(\mathbf{X}, \mathbf{x}_\star)^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{X}, \mathbf{z}_\star)$$

where we write $\mathbf{k}(\mathbf{X}, \mathbf{x}_\star)$ to denote a vector of covariance evaluations.

[2]Standard manipulations of the Gaussian distribution are given in appendix A.

The process of learning can be identified with the matrix inversion in mean and covariance predictions, an operation which scales with N^3. After inverting $(\mathbf{K}+\sigma^2\mathbf{I})$, we can evaluate the mean at new test inputs in time $\mathcal{O}(N)$ and the variance in $\mathcal{O}(N^2)$. The transition from prior to posterior process in the presence of observations is illustrated in fig. 1.2b.

1.2.3 Model selection in Gaussian processes

We find the log marginal likelihood by integrating out the latent $\mathbf{f}$:

$$\begin{aligned}\log p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta}) &= \log \int \mathcal{N}\big(\mathbf{y}\,;\,\mathbf{f}\,,\,\sigma^2\mathbf{I}\big)\,\mathcal{N}(\mathbf{f}\,;\,\mathbf{0}\,,\,\mathbf{K_{ff}})\,\mathrm{d}\mathbf{f} \\ &= -\frac{1}{2}\mathbf{y}^T(\mathbf{K_{ff}}+\sigma^2\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K_{ff}}+\sigma^2\mathbf{I}| - \frac{N}{2}\log 2\pi, \qquad (1.5)\end{aligned}$$

where $\boldsymbol{\theta}$ again denotes hyperparameters, i.e. parameters of the kernel function. There are three terms: the first involves the observed targets $\mathbf{y}$ and penalizes poor fit; the second is independent of the targets and penalizes complexity; the third is a normalization constant. By changing properties of the kernel such as lengthscale and variance, we can encourage the model to favour a priori certain data sets over others. The method of type-II ML is to set the hyperparameters to those of the model most likely to have generated the observations.

It is testament to the power of the Bayesian framework that we are able to make non-trivial statements about hyperparameters based on just two data points: in fig. 1.3 are evidence contours for a variety of GP models using the squared exponential kernel, after two (one-dimensional) observations, denoted by the black dot (t_1, t_2). For large l, the model expects t_1 and t_2 to be identical; when this is not the case it is strongly penalized. Conversely, for small l, we would expect the observations to be independent and there is "surprise" if they are not: predictive mass over function space is wasted in regions where the outputs would appear independent. We conclude that the optimum must occur at some intermediate value. The signal variance is regulated in a similar way: very small values for a concentrate probability mass close to the origin and cannot explain a significant deviation of the observations from zero; very large values flatten the Gaussian pancake over an ever wider area whose value at the observation (t_1, t_2) must eventually tend towards zero. In consequence, the data support the belief in intermediate values both for lengthscale and for magnitude.

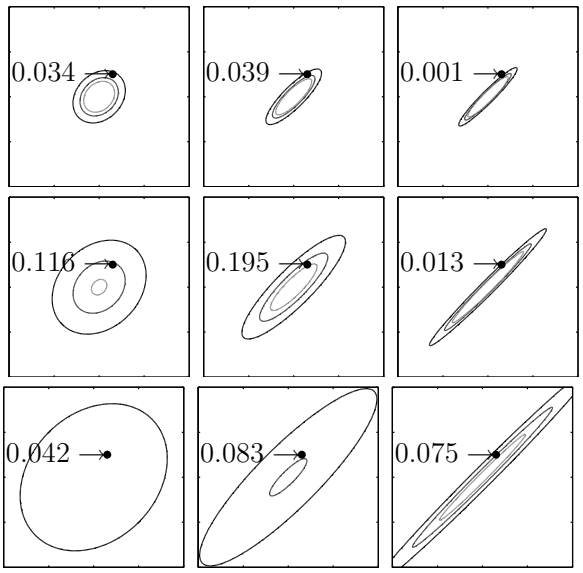


Figure 1.3: These plots illustrate model selection on a training set of size two. Evidence contours are shown for the squared exponential kernel (1.3) under a variety of parameterizations; we seek to maximize the value of the probability density at the data (marked by a black dot), as a function of the process lengthscale and signal variance. The central image shows the optimal model, while the surrounding images show how extremes of lengthscale (too short on the left, too long on the right) and variance (too small on the top, too large on the bottom) are penalized.

Any locally optimal setting for hyperparameters may be understood as a feasible description of the underlying generative process. However, real datasets do not always exhibit a unique optimum, as illustrated in fig. 1.4. How we favour one model over another will depend on the extent to which we embrace the Bayesian paradigm: strictly, we should always integrate out the hyperparameters, whose posterior distribution is

$$p(\boldsymbol{\theta}|\mathbf{X},\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X})}{\int p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X})\mathrm{d}\boldsymbol{\theta}}, \tag{1.6}$$

where $p(\boldsymbol{\theta}|\mathbf{X})$ is a hyperprior, that is, a prior on the hyperparameters. The marginal likelihood from (1.5) appears here as the likelihood of the data given $\boldsymbol{\theta}$: although a normalized Gaussian in $\mathbf{y}$, its distribution over $\boldsymbol{\theta}$ is usually very complicated, making the integral in (1.6) analytically intractable. The committed Bayesian must then resort to Monte Carlo methods (see section 1.5).

Type-II ML provides a much faster alternative: provided the class of models is appropriate, we can expect with sufficient training data that the global optimum in $\boldsymbol{\theta}$ of the marginal likelihood will be significantly more plausible than competing interpre-

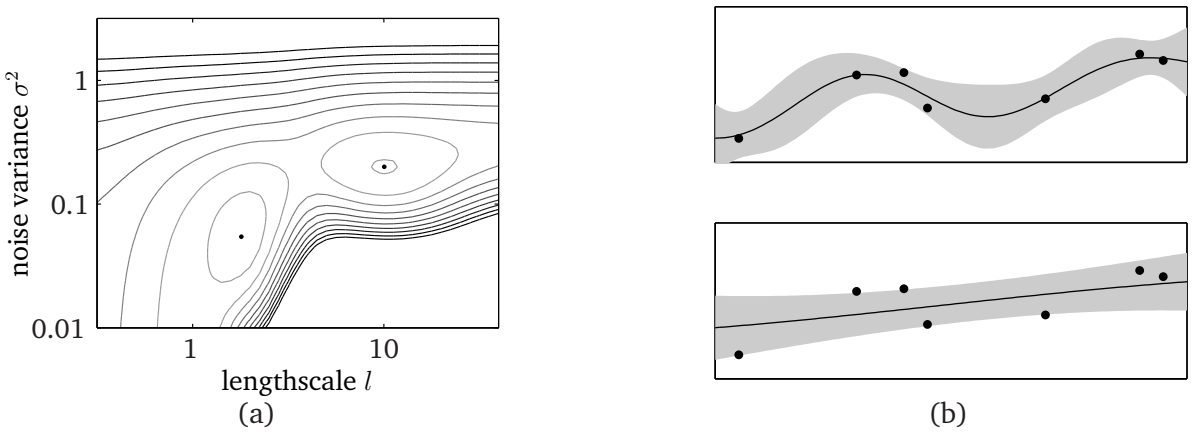


Figure 1.4: Data sets can accommodate several explanations, appearing as local maxima in the marginal likelihood. In panel (a) we see two peaks in the evidence which correspond to the alternative explanations of panel (b): the former a higher-frequency function with lower noise variance, the latter a noisier, near-linear signal.

tations. A local maximum $\hat{\boldsymbol{\theta}}$ can be found by gradient ascent on (1.5) and this single model used to make predictions. The approximation will be adequate provided we indeed locate the global optimum and that it is sufficiently well-peaked to be approximated by a delta function; for more details, see MacKay (1999). Notice that since our prior on $\mathbf{f}$ is always fixed at zero-mean and the kernel usually has very few parameters, we very rarely witness the overfitting problems that plague traditional ML techniques. However, there is no certainty that a model favoured by the evidence will be a good predictor on fresh data. If the kernel is inappropriate for the task (e.g. a linear kernel fitting periodic observations), no predictor will generalize well: this is a question of model mismatch. Problems with overfitting can also occur if a highly flexible kernel is employed with too small a training set. Difficulties of this nature can in general be detected only by using a held-out set for validation; in practice, the efficiency and success of evidence maximization make type-II ML a popular choice.

Maximizing the marginal likelihood of a GP model with an anisotropic kernel achieves *automatic relevance determination*, a probabilistically well-founded approach to identifying salient features (Neal, 1996): if our observations indicate that the output of a function is fixed or varies only very slowly with respect to certain inputs, the most plausible generative model will have lengthscales on these components which are very large, and hence the associated inputs may to a first approximation be ignored.

1.2.4 Gaussian processes as linear models

Gaussian processes can also be recovered from a Bayesian perspective on linear models, where in general the correspondence applies only for an infinite number of components. Let the M basis functions $\phi_m(\cdot)$ of a linear model operate on the data $\mathbf{X}$ to form a design matrix $\mathbf{\Phi}$ of dimensions $N \times M$, where $\Phi_{nm} = \phi_m(\mathbf{x}_n)$. Consider a linear combination of these bases weighted by a vector $\mathbf{w} = \{w_m\}_{m=1}^M$, so that

$$f(\mathbf{x}_n) = \sum_{m=1}^{M} w_m \phi_m(\mathbf{x}_n) \qquad \Longrightarrow \qquad \mathbf{f} = \mathbf{\Phi}\mathbf{w}.$$

If we place on $\mathbf{w}$ a Gaussian prior $\mathcal{N}(\mathbf{w}\,;\,\mathbf{0}\,,\,\mathbf{A})$ then $p(\mathbf{f}) = \mathcal{N}\big(\mathbf{f}\,;\,\mathbf{0}\,,\,\mathbf{\Phi}\mathbf{A}\mathbf{\Phi}^T\big)$, which is recognised as a GP prior on $\mathbf{f}$ with zero mean and covariance function

$$k_{\mathbf{A}}(\mathbf{x}, \mathbf{z}) = \boldsymbol{\phi}^T(\mathbf{x})\mathbf{A}\boldsymbol{\phi}(\mathbf{z}). \tag{1.7}$$

From here it is possible to introduce a noise model $p(\mathbf{y}|\mathbf{f})$ and invoke Bayes' theorem to obtain a posterior on $\mathbf{f}$, and thence to make predictions; see for example O'Hagan and Forster (2004, ch. 9). Indeed, this is computationally a worthwhile approach when $M < N$ since inference costs in the linear model scale with M^3.

Returning to (1.7), it is clear that covariance matrices formed from this function can never have rank greater than M. By virtue of Mercer's theorem (König, 1986), we can theoretically express *any* kernel function as a potentially infinite sum in terms of its eigenfunctions and eigenvalues

$$k(\mathbf{x}, \mathbf{z}) = \sum_{m=1}^{\infty} \lambda_m \psi_m(\mathbf{x}) \psi_m(\mathbf{z}).$$

When the number of terms is finite, such as in the explicit linear model described above, the kernel is said to be *degenerate*. Models derived from such kernels can exhibit counterintuitive behaviour; see for example Rasmussen and Quiñonero-Candela (2005) and Quiñonero-Candela et al. (2007), and section 2.1.4. If there are an infinite number of terms, such as for the squared exponential and most other commonly-used kernels, the derived covariance matrices are always full rank, and the kernel is non-degenerate. Quiñonero-Candela (2004, sec. 3.2) discusses the relationship between GPs and linear models in greater detail.

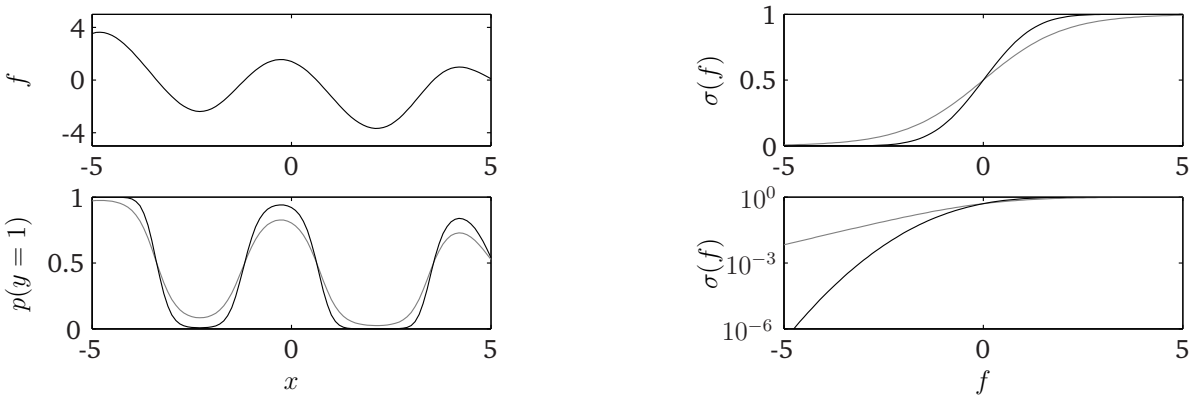


Figure 1.5: The standard model for GP classification uses a sigmoidal function to transform a real latent signal (top left) into one constrained to $[0, 1]$ (bottom left). Two such functions are illustrated: the feint line corresponds to the logit, the black line to the probit. Observe that the former is more conservative in its assignment of predictive probabilities.

1.2.5 Gaussian process classification

With the introduction of an appropriate noise model, GPs can be readily applied to classification problems. We typically assume that examples are labelled $y_n \in \{\pm 1\}$ in a probabilistic manner, by passing the latent f_n through a sigmoidal function bounded by $[0, 1]$, to obtain the probability $p(y_n = +1|f_n)$. Examples of such functions are the logit and probit transforms, illustrated in fig. 2.2a, and defined respectively

$$\lambda(f_n) = \frac{1}{1 + \exp(-f_n)}, \qquad \sigma(f_n) = \int_{-\infty}^{f_n} \mathcal{N}(z\,;\,0\,,\,1)\,\mathrm{d}z.$$

In both cases the function is antisymmetric so the probability $p(y_n = -1)$ of the opposite label is $1 - \sigma(f_n) = \sigma(-f_n)$, hence $p(y_n) = \sigma(y_n f_n)$.[3] Both have been used in GP classification: the logit is employed by Gibbs and MacKay (2000) in a variational approximation (see section 1.4) where it is bounded above and below by an exponential; use of the probit is more common for an approximation based on moment matching (see section 1.3) since it renders the marginal distribution $\int p(y_n|f_n)p(f_n)\mathrm{d}f_n$ and hence moments of f_n analytically tractable.

We assume the same zero-mean Gaussian prior over $\mathbf{f}$ as before:

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}\,;\,\mathbf{0}\,,\,\mathbf{K_{ff}})\,.$$

[3] We use the function symbol $\sigma(\cdot)$ to refer to a sigmoid function from now on, since this thesis will be concerned almost exclusively with the probit.

The likelihood factorizes into a product of N terms

$$p(\mathbf{y}|\mathbf{f}) = \prod_{n=1}^{N} p(y_n|f_n) = \prod_{n=1}^{N} \sigma(y_n f_n),$$

from which Bayes' rule gives the posterior distribution over $\mathbf{f}$:

$$p(\mathbf{f}|X, \mathbf{y}) = \frac{p(\mathbf{f}|\mathbf{X}) \prod_{n=1}^{N} \sigma(y_n f_n)}{\int p(\mathbf{f}|\mathbf{X}) \prod_{n=1}^{N} \sigma(y_n f_n) \mathrm{d}\mathbf{f}}. \tag{1.8}$$

Predictions at $\mathbf{x}_\star$ are made by marginalizing over the latent $f_\star$:

$$p(f_\star|\mathbf{X}, \mathbf{y}, \mathbf{x}_\star) = \int p(f_\star|\mathbf{f}, \mathbf{X}, \mathbf{x}_\star) p(\mathbf{f}|\mathbf{X}, \mathbf{y}) \mathrm{d}\mathbf{f};$$
$$p(\mathbf{y}_\star = +1|\mathbf{X}, \mathbf{y}, \mathbf{x}_\star) = \int \sigma(f_\star) p(f_\star|\mathbf{X}, \mathbf{y}, \mathbf{x}_\star) \mathrm{d}f_\star.$$

The non-Gaussian likelihood $\sigma(\cdot)$ introduces complications in the evaluation of the posterior (1.8): latent function values of the wrong sign are strongly penalized by the soft threshold of the likelihood, while through the prior we penalize to a lesser extent latent values of large magnitude. The posterior in consequence is *asymmetric*: it falls sharply in regions rejected by the data, and gently in regions discouraged by the prior. The integral needed to normalize the joint distribution is in fact intractable, and we turn now to methods of approximation.

1.3 Expectation propagation

Bayesian inference invariably requires the solution of integrals involving the posterior, through marginalization for expectations and predictive distributions, or in calculating the evidence. In certain cases, such as the GP with a Gaussian noise model, the requisite integrals have a mathematically closed form; in most other cases they do not, or its evaluation is of exponential complexity. We are forced then to use either a tractable but inexact form for the posterior which does allow integration, or to draw sufficient independent samples from it by a Monte Carlo algorithm that a stochastic average constitutes a reasonable approximation to the true solution.

In the GP, we regain the advantage of tractable evidence and predictive distributions by approximating the full posterior with a single multivariate Gaussian. The classic technique based on this idea is Laplace's method, which fits the mean at the peak

of the posterior and matches the curvature there. The approximation is symmetric and local which can lead to very inaccurate estimates when the true distribution is strongly asymmetric, as we have suggested will be the case for binary classification. Only by considering global properties of the approximated distribution can we expect significant improvements, and in this thesis we will largely be concerned with the expectation propagation (EP) algorithm of Minka (2001), which attempts to match every marginal moment of the full posterior distribution.

Consider an intractable distribution over $\mathbf{u}$ that factorizes into a product of terms, for example a prior distribution $t_0(\mathbf{u})$ and a series of likelihoods $\{t_n(y_n|\mathbf{u})\}_{n=1}^N$:

$$p(\mathbf{u}|\mathbf{y}) \propto t_0(\mathbf{u}) \prod_{n=1}^{N} t_n(y_n|\mathbf{u}). \tag{1.9}$$

EP constructs an approximation to (1.9) as a product of scaled *site functions* $\tilde{t}_n(\mathbf{u})$. For computational efficiency and tractability, these sites are usually chosen from an exponential family (see appendix A) with natural parameters $\boldsymbol{\theta}$, since in this case the product

$$q(\mathbf{u};\boldsymbol{\theta}) = \prod_{n=0}^{N} \tilde{t}_n(\mathbf{u};\boldsymbol{\theta}_n)$$

retains the same functional form as its components; that is, q is also a member of the same exponential family. The optimal solution would be a set of parameters $\boldsymbol{\theta}$ that minimizes some global measure, for example the Kullback-Leibler divergence

$$\min_{\boldsymbol{\theta}} \mathsf{KL}\big(p(\mathbf{u}|\mathbf{y}) \| q(\mathbf{u};\boldsymbol{\theta})\big),$$

but this optimization is generally intractable. EP is an iterative procedure that uses the same divergence measure, but refines its approximation on a termwise basis only. At each iteration, a new site n is selected. The product of the *cavity* distribution,

$$q^{\backslash n}(\mathbf{u};\boldsymbol{\theta}^{\backslash n}) = \prod_{n' \neq n} \tilde{t}_{n'}(\mathbf{u};\theta_{n'}),$$

formed by the current approximation with the omission of that site, and the true likelihood term t_n, yields the *tilted* distribution

$$q^{n}(\mathbf{u};\boldsymbol{\theta}^{\backslash n}) = t_n(y_n|\mathbf{u}) \prod_{n' \neq n} \tilde{t}_{n'}(\mathbf{u};\theta_{n'}). \tag{1.10}$$

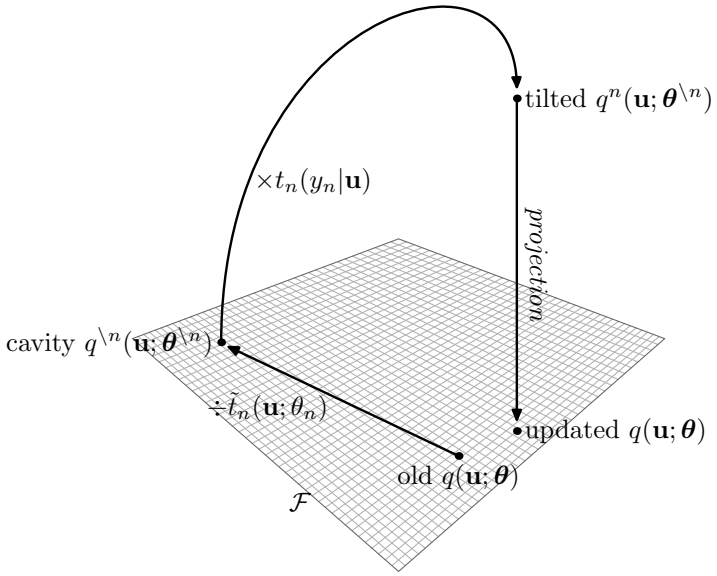


Figure 1.6: Each iteration of expectation propagation can be seen as projecting the tilted distribution back into the function space $\mathcal{F}$ such that the moments are preserved.

The simpler optimization

$$\min_{\theta_n} \mathsf{KL}\left(q^n(\mathbf{u};\boldsymbol{\theta}^{\setminus n}) \| q(\mathbf{u};\boldsymbol{\theta}^{\setminus n},\theta_n)\right)$$

is then performed, fitting only the parameters θ_n. Using this divergence can be shown equivalent to matching the moments of the two *marginal* distributions—although there is no guarantee that the global moments will also match. The minimization can also be understood as finding an optimal projection for the tilted distribution (1.10) that lies within the chosen exponential family (see fig. 1.6). After each site update, the moments at the remaining sites are liable to change, and several iterations may be required before convergence.

1.3.1 EP for natural Gaussian site functions

Considerable simplifications occur if the sites are Gaussians under natural parameterization (see appendix A), each of which refers only to a subset J_n of variables:

$$\tilde{t}_n(\mathbf{u}_{J_n}) = \mathcal{N}^U(\mathbf{u}_{J_n}\,;\,\mathbf{b}_n\,,\,\boldsymbol{\Pi}_n)\,.$$

Let the prior be Gaussian, such that $t_0 = \tilde{t}_0 = \mathcal{N}(\mathbf{u}\,;\,\mathbf{h}_0\,,\,\mathbf{A}_0)$, and let the approximate posterior be $\mathcal{N}(\mathbf{u}\,;\,\mathbf{h}\,,\,\mathbf{A})$. The approximation is initialized equal to the prior and all

site parameters set to zero, before steps (1) to (4) below are iterated for every site. After each full cycle, the posterior is refreshed using (5).

1. The marginal posterior $Q(\mathbf{u}_{J_n}) = \mathcal{N}(\mathbf{u}_{J_n}\,;\,\mathbf{h}_{J_n}\,,\,\mathbf{A}_{J_n})$ is calculated, from which we find the cavity distribution $Q^{\backslash n}(\mathbf{u}_{J_n}) = \mathcal{N}\Big(\mathbf{u}_{J_n}\,;\,\mathbf{h}^{\backslash n}_{J_n}\,,\,\mathbf{A}^{\backslash n}_{J_n}\Big)$:

$$\mathbf{A}^{\backslash n}_{J_n} = \mathbf{A}_{J_n}(\mathbf{I} - \mathbf{A}_{J_n}\mathbf{\Pi}_n)^{-1}, \qquad \mathbf{h}^{\backslash n}_{J_n} = \mathbf{h}_{J_n} + \mathbf{A}^{\backslash n}_{J_n}(\mathbf{\Pi}_n\mathbf{h}_{J_n} - \mathbf{b}_n). \tag{1.11}$$

2. Given an analytic expression for the zeroth moments Z_n of the tilted Gaussian $Q^{\backslash n}(\mathbf{u}_{J_n})t_n(\mathbf{u}_{J_n})$, we obtain derivatives w.r.t $\mathbf{h}^{\backslash n}_{J_n}$:

$$Z_n = \int Q^{\backslash n}(\mathbf{u}_{J_n})t_n(\mathbf{u}_{J_n})\mathrm{d}\mathbf{u}_{J_n}, \quad \boldsymbol{\alpha}_n = \nabla_{\mathbf{h}^{\backslash n}_{J_n}} \log Z_n, \quad \boldsymbol{\nu}_n = -\nabla^2_{\mathbf{h}^{\backslash n}_{J_n}} \log Z_n. \tag{1.12}$$

   Observe that if $Z_n = Z^R_n + Z^O_n$, as in chapter 3, then

$$\boldsymbol{\alpha}_n = \frac{1}{Z_n}\left(\nabla_{\mathbf{h}^{\backslash n}_{J_n}} Z^R_n + \nabla_{\mathbf{h}^{\backslash n}_{J_n}} Z^O_n\right), \quad \boldsymbol{\nu}_n = \boldsymbol{\alpha}_n\boldsymbol{\alpha}_n^T - \frac{1}{Z_n}\left(\nabla^2_{\mathbf{h}^{\backslash n}_{J_n}} Z^R_n + \nabla^2_{\mathbf{h}^{\backslash n}_{J_n}} Z^O_n\right).$$

3. The new marginal posterior distribution has moments

$$\mathbf{h}'_{J_n} = \mathbf{h}^{\backslash n}_{J_n} + \mathbf{A}^{\backslash n}_{J_n}\boldsymbol{\alpha}_n, \qquad \mathbf{A}'_{J_n} = (\mathbf{I} - \mathbf{A}^{\backslash n}_{J_n}\boldsymbol{\nu}_n)\mathbf{A}^{\backslash n}_{J_n}.$$

   These can be used in the Woodbury formula to calculate the new site parameters directly:

$$\mathbf{\Pi}'_n = \boldsymbol{\nu}_n\left(\mathbf{I} - \mathbf{A}^{\backslash n}_{J_n}\boldsymbol{\nu}_n\right)^{-1}, \qquad \mathbf{b}'_n = \mathbf{\Pi}'_n\left(\mathbf{h}^{\backslash n}_{J_n} + \boldsymbol{\nu}_n^{-1}\boldsymbol{\alpha}_n\right). \tag{1.13}$$

4. A rank-J_n update is made to the posterior. Let the change in precision be $\Delta_n = \mathbf{\Pi}'_n - \mathbf{\Pi}_n$, and write $\mathbf{a}_{J_n}$ for the J_n columns of $\mathbf{A}$. Using the Woodbury formula,

$$\mathbf{A}' = \mathbf{A} - \mathbf{a}_{J_n}\left(\mathbf{A}_{J_n} + \Delta_n^{-1}\right)^{-1}\mathbf{a}^T_{J_n}, \qquad \mathbf{h}' = \mathbf{A}'\left(\mathbf{A}_0^{-1}\mathbf{h}_0 + \mathbf{b}\right). \tag{1.14}$$

5. Repeated low-rank updates cause loss of precision. Occasionally, we must refresh the posterior from the prior using all the site parameters $(\mathbf{\Pi}, \mathbf{b})$:

$$\mathbf{A} := \left(\mathbf{A}_0^{-1} + \mathbf{\Pi}\right)^{-1}, \qquad \mathbf{h} := \mathbf{A}\left(\mathbf{A}_0^{-1}\mathbf{h}_0 + \mathbf{b}\right). \tag{1.15}$$

1.3.2 Marginal likelihood approximation

EP provides also an estimate of the log *marginal likelihood*

$$\log p(\mathbf{y}) = \log \int_{\mathbf{u}} t_0(\mathbf{u}) \prod_{n=1}^{N} t_n(y_n|\mathbf{u}) \mathrm{d}\mathbf{u}$$

by explicitly matching the zeroth-order moments, or scale, at each site inclusion. If the posterior approximation is from an exponential family $\mathcal{F}$, Seeger (2005) shows the estimated marginal likelihood to be

$$L = \sum_{n=1}^{N} \log C_n + \Phi(\boldsymbol{\theta}) - \Phi(\boldsymbol{\theta}^{(0)}), \quad \text{where} \quad \log C_n = \log Z_n - \Phi(\boldsymbol{\theta}) + \Phi(\boldsymbol{\theta}^{\backslash n}); \quad (1.16)$$

$\Phi(\cdot)$ denotes the log partition function for $\mathcal{F}$ and $\boldsymbol{\theta}^{(0)}$ are the natural parameters of the prior.

For model selection, we also require derivatives of the marginal likelihood with respect to hyperparameters $\boldsymbol{\xi}$. Seeger (2005) establishes the following expressions:

$$\nabla_{\boldsymbol{\xi}^{(0)}} L = \mathsf{Tr}\left(\left(\boldsymbol{\eta} - \boldsymbol{\eta}^{(0)}\right) \nabla_{\boldsymbol{\xi}^{(0)}} \boldsymbol{\theta}^{(0)}\right) \qquad \text{and} \qquad \nabla_{\boldsymbol{\xi}^n} L = \nabla_{\boldsymbol{\xi}^{(n)}} \log Z_n, \qquad (1.17)$$

where $\boldsymbol{\xi}^{(0)}$ are hyperparameters of the prior such as kernel lengthscales etc., $\boldsymbol{\xi}^{(n)}$ are hyperparameters of site n only, and $\boldsymbol{\eta}$ denotes the moment parameters of $\mathcal{F}$. In other words, by taking advantage of the fixed point conditions of EP, which establish the consistency of true and approximate moments up to second order of all the marginal distributions, the dependencies between L and sites not directly affected by changes in $\boldsymbol{\xi}^{(0)}$ and $\boldsymbol{\xi}^{(n)}$ cancel; see Seeger (2005) for details.

1.4 Variational methods

Variational methods allow a complex inference problem to be exchanged for a simpler approximation parameterized by an associated set of *variational parameters*. The approximation forms a strict bound on the marginal likelihood of the original model, and the task of learning becomes one of finding an optimal setting for these extra variables to make the bound as tight as possible. In the context of machine learning, the ideas were presented first by Hinton and van Camp (1993), although the core concepts had appeared much earlier in the field of statistical physics (Feynman, 1972).

The framework for variational methods in models with hidden or *latent* variables can be derived from expectation maximization (EM), a classical method for parameter optimization which we now review.[4] Consider a model with parameters $\boldsymbol{\theta}$, and where each observation has associated latent variable(s) u_n: if we make i.i.d. observations $\mathbf{y} = \{y_n\}_{n=1}^N$, the evidence is given by marginalizing over all u_n in the joint distribution (where both y_n and u_n may be vectors). By an application of Jensen's inequality, we find that for any auxiliary distributions $\mathbf{q}(\mathbf{u}) = \{q_n(u_n)\}_{n=1}^N$, the log marginal likelihood

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &\doteq \log \int p(\mathbf{y}, \mathbf{u}|\boldsymbol{\theta}) \mathrm{d}\mathbf{u} \\
&= \sum_{n=1}^N \log \int q_n(u_n) \frac{p(y_n, u_n|\boldsymbol{\theta})}{q_n(u_n)} \mathrm{d}u_n \\
&\geq \sum_{n=1}^N \int q_n(u_n) \log \left(\frac{p(y_n, u_n|\boldsymbol{\theta})}{q_n(u_n)} \right) \mathrm{d}u_n \doteq \mathcal{F}(\mathbf{q}(\mathbf{u}), \boldsymbol{\theta}) \\
&= \mathcal{L}(\boldsymbol{\theta}) - \sum_{n=1}^N \int q_n(u_n) \log \left(\frac{q_n(u_n)}{p(u_n|y_n, \boldsymbol{\theta})} \right) \mathrm{d}u_n,
\end{aligned}$$

where $\mathcal{F}(\mathbf{q}(\mathbf{u}), \boldsymbol{\theta})$ is a lower bound on $\mathcal{L}(\boldsymbol{\theta})$ and achieves equality only when $q_n(u_n) = p(u_n|y_n, \boldsymbol{\theta})$, evident from the final term which is the (non-negative) KL-divergence between q and p. If we impose restrictions on the form of q (e.g. factorizing over elements of u_n when it is vectorial), this divergence will be a positive value for any distribution $q_n(u_n)$ if the true distribution on u_n does not also satisfy the restrictions.

The EM algorithm is a maximum likelihood procedure for fitting parameters $\boldsymbol{\theta}$ by alternately optimizing the bound $\mathcal{F}$ as a function of $\boldsymbol{\theta}$ for fixed statistics $q(\mathbf{u})$ (the M-step), and inference of the distribution over $\mathbf{u}$ for given $\boldsymbol{\theta}$ (the E-step). If the distributions $q_n(u_n)$ are unconstrained, the E-step makes the bound tight; if each q_n is from a restricted family then optimization of its variational parameters $\boldsymbol{\lambda}_n$ yields the "variational EM" algorithm. The procedure is guaranteed to converge, and in the exact case finds a local optimum of $\mathcal{L}$. However, in common with other maximum likelihood methods, EM concentrates on the *density* of the posterior over model parameters, rather than on its mass—a fully Bayesian treatment would also accommodate a distribution on parameters, and this is what variational Bayes or "ensemble methods" achieve.

[4] Variational methods are more widely applicable: see Jordan et al. (1999) or Beal (2003) for a broader introduction and various applications.

We extend the methodology of EM by introducing a distribution over $\boldsymbol{\theta}$ and imposing a factorization constraint on the auxiliary distribution $q(\mathbf{u}, \boldsymbol{\theta}) \approx \mathbf{q}_{\mathbf{u}}(\mathbf{u}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$. This yields a bound on the evidence similar to that seen above;

$$\begin{aligned}
\mathcal{L} &\doteq \log \int p(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}) \mathrm{d}\mathbf{u}\mathrm{d}\boldsymbol{\theta} \\
&= \sum_{n=1}^{N} \log \int q_n(u_n) q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \frac{p(y_n, u_n, \boldsymbol{\theta})}{q_n(u_n) q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \mathrm{d}\mathbf{u}\mathrm{d}\boldsymbol{\theta} \\
&\geq \sum_{n=1}^{N} \int q_n(u_n) q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \log \left(\frac{p(y_n, u_n, \boldsymbol{\theta})}{q_n(u_n) q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \right) \mathrm{d}\mathbf{u}\mathrm{d}\boldsymbol{\theta} \doteq \mathcal{F}(\mathbf{q}_{\mathbf{u}}(\mathbf{u}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})).
\end{aligned}$$

Now equivalent E- and M-steps are alternated for the distributions over hidden variables and parameters, which by functional differentiation are discovered to be

$$\begin{aligned}
q_n^{(t+1)}(u_n) &\propto \exp \left(\int q_{\boldsymbol{\theta}}^{(t)}(\boldsymbol{\theta}) \log p(y_n, u_n | \boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \right) && \text{VBE-step;} \\
q_{\boldsymbol{\theta}}^{(t+1)}(\boldsymbol{\theta}) &\propto p(\boldsymbol{\theta}) \exp \left(\int q_{\mathbf{u}}^{(t+1)}(\mathbf{u}) \log p(\mathbf{y}, \mathbf{u} | \boldsymbol{\theta}) \mathrm{d}\mathbf{u} \right) && \text{VBM-step,}
\end{aligned}$$

where $q_{\mathbf{u}}(\mathbf{u}) = \prod_{n=1}^{N} q_n(u_n)$, a factorization which occurs as a consequence of the i.i.d. assumption. We see that whereas the original EM optimized in its M-step a bound on the evidence by selecting optimal $\boldsymbol{\theta}$, in variational methods the evidence $\mathcal{L}$ is a constant since both $\mathbf{u}$ and $\boldsymbol{\theta}$ are marginalized; the associated M-step only ever improves the *bound* $\mathcal{F}$, thus incorporating a natural complexity penalty where standard EM can "blow up". This bound may also be used sensibly as a guide for model selection.

Recall that the KL divergence is asymmetric. In variational methods we use $\mathrm{KL}(q \| p) = \int q(\mathbf{u}) \log \frac{q(\mathbf{u})}{p(\mathbf{u})} \mathrm{d}\mathbf{u}$, in which the expectation is with respect to the approximating distribution. In a sense, this is the "wrong way around", because the region in which the approximation must be accurate is governed not by the true distribution but by the approximation itself. The result is a consistent *underestimation* of variance: we are penalized for placing excessive q-mass in regions where p has little, and tend to focus on a single mode of the posterior. In contrast, the divergence employed by EP, in which the arguments are reversed, demands that q is somehow representative across the entire space of p; it penalizes distributions which *do not* have mass where p places significant probability. In fact, both forms of the divergence are two points on a continuous scale of α-divergences (Amari, 1985), which interpolates smoothly between (and beyond) the two regimes; see also Paquet (2007).

1.5 Stochastic inference

Instead of approximating a complicated distribution $p(\mathbf{x})$ with something tractable $q(\mathbf{x})$ and using the approximation to evaluate expectations

$$\langle f(\mathbf{x})\rangle_{p(\mathbf{x})} = \int f(\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x} \approx \int f(\mathbf{x})q(\mathbf{x})\mathrm{d}\mathbf{x},$$

we can attempt to generate independent samples $\mathbf{x}^{(t)}$ from p, such that almost surely by the strong law of large numbers, as $T \to \infty$, averages with respect to the samples converge to the true result:

$$\int f(\mathbf{x})p(\mathbf{x})\mathrm{d}x \approx \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{x}^{(t)}). \tag{1.18}$$

The challenge is then to devise efficient means of generating effectively independent samples from arbitrary distributions. In this field the literature is very extensive, and we omit discussion of certain methods (importance sampling and rejection sampling, for example) that are not directly relevant to the thesis. A more expansive review appears in MacKay (2003, ch. 29 and ch. 30).

1.5.1 Markov chains

The fundamental aim of Markov chain Monte Carlo (MCMC) is to establish a Markov chain that, in the limit of infinite time, draws independent samples from the posterior distribution of interest. Formally, a first-order Markov chain consists of a series of random variables $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$ with the conditional independence assumption

$$t(\mathbf{x}^{(\tau+1)}|\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(\tau)}) = t(\mathbf{x}^{(\tau+1)}|\mathbf{x}^{(\tau)}).$$

We will consider only homogeneous Markov chains, for which transition probabilities $t(\mathbf{x}^{(\tau+1)}|\mathbf{x}^{(\tau)})$ are the same for all τ. A distribution $s(\mathbf{x})$ is *invariant* with respect to a set of transition probabilities provided

$$s(\mathbf{x}) = \sum_{\mathbf{x}'} t(\mathbf{x}|\mathbf{x}')s(\mathbf{x}');$$

a sufficient condition for invariance is that the transitions satisfy *detailed balance*:

$$s(\mathbf{x})t(\mathbf{x}'|\mathbf{x}) = s(\mathbf{x}')t(\mathbf{x}|\mathbf{x}').$$

We will also require *ergodicity*, i.e. that for any initial distribution $t(\mathbf{x}^{(0)})$, the chain eventually converges to the invariant or equilibrium distribution. *Aperiodicity* and *irreducibility* are sufficient to ensure this property of homogeneous Markov chains: the first condition means there are no cycles of states (for which there could not be convergence); the second that it must be possible to reach any state from any other, i.e. the transition graph is connected. For further details, consult Neal (1993) or Andrieu et al. (2003). Note that, although consecutive states of the chain are correlated, sufficiently well-separated samples can be considered essentially independent.

1.5.2 The Metropolis-Hastings algorithm

In order to establish the desired invariance, we employ the following selection criterion. From state $\mathbf{x}^{(\tau)}$ we make a *proposal* by drawing a sample $\mathbf{x}_\star$ from the proposal distribution $q(\mathbf{x}_\star|\mathbf{x}^{(\tau)})$; this may be any potentially asymmetric distribution, but is most commonly the isotropic Gaussian. The probability of acceptance is given by

$$a(\mathbf{x}_\star|\mathbf{x}^{(\tau)}) = \min\left(1, \frac{P(\mathbf{x}_\star)}{P(\mathbf{x}^{(\tau)})}\frac{q(\mathbf{x}^{(\tau)}|\mathbf{x}_\star)}{q(\mathbf{x}_\star|\mathbf{x}^{(\tau)})}\right), \tag{1.19}$$

where the normalized posterior $p(\mathbf{x}) = \frac{1}{Z_p}P(\mathbf{x})$, but evaluation of the normalizing constant Z_p is not required. The second factor corrects for bias in the transition probabilities due to an asymmetric q. This criterion appeared first in Hastings (1970), generalizing the work of Metropolis et al. (1953) which was applicable only to symmetric proposals. If the new state is accepted, we update $\mathbf{x}^{(\tau+1)} := \mathbf{x}_\star$; if the new state is rejected, the current state is repeated $\mathbf{x}^{(\tau+1)} := \mathbf{x}^{(\tau)}$.

It is straightforward to show that this procedure satisfies the demand of detailed balance, and hence that the posterior $p(\mathbf{x})$ is the invariant distribution of the chain. The problem with a naïve application of Metropolis-Hastings is the rate at which it explores the state space. If we consider the isotropic Gaussian proposal, at each step its suggested new state $\mathbf{x}_\star$ is independent of previous choices, giving rise to *random walk* behaviour where in general it takes a number of steps proportional to N^2 to travel a distance N from our initial state. We can reduce the constant factor by making larger jumps (i.e. broadening the proposal distribution), but only at the risk of a greater rejection rate: consider an elongated Gaussian from which we seek to draw samples, whose largest component of variance is σ^2_{max}, and whose smallest is σ^2_{min}. An independent sample requires that we traverse its longest dimension, but to keep rejection rates

low, the proposal density should not be much broader than σ^2_{min}, hence we will need to draw $\mathcal{O}\left((\sigma^2_{\text{max}}/\sigma^2_{\text{min}})^2\right)$ intermediate samples.

From the initial state $\mathbf{x}^{(0)}$ there will be some time during which the chain is migrating towards the typical set of states; this period is called the *burn-in* and should be discarded from the set of samples used in the approximation (1.18). As mentioned above, states $\mathbf{x}^{(\tau)}$ and $\mathbf{x}^{(\tau+1)}$ are strongly correlated, and to accumulate independent draws from the posterior, we may have to discard many of these intermediate samples. To help identify these periods, it is useful to examine the auto-correlation of the chain for different displacements.

Gibbs sampling

A special case of Metropolis-Hastings is Gibbs sampling, in which we update the state componentwise, drawing samples from the relevant conditional distributions. After initializing $\mathbf{x}^{(0)} = \{x_1^{(0)}, x_2^{(0)}, \ldots, x_D^{(0)}\}$,

$$\begin{aligned}
&\text{sample } x_1^{(t+1)} \text{ from } p(x_1|x_2^{(t)}, \ldots, x_D^{(t)}) \\
&\text{sample } x_2^{(t+1)} \text{ from } p(x_2|x_1^{(t+1)}, x_3^{(t)}, \ldots, x_D^{(t)}) \\
&\qquad\qquad \vdots \\
&\text{sample } x_D^{(t+1)} \text{ from } p(x_D|x_1^{(t+1)}, \ldots, x_{D-1}^{(t+1)}).
\end{aligned}$$

Two benefits are that we no longer need devise a suitable proposal distribution, and that states are always accepted (it can be confirmed that the acceptance ratio (1.19) will always evaluate to 1). However, we do not eliminate random walk behaviour, and it becomes difficult to achieve coordinated updates in the state since by construction it is allowed to evolve only by component-wise transitions. An idea called *overrelaxation* (Adler, 1981; Neal, 1995) makes some compensation by biasing the updates conditional on the current $\mathbf{x}^{(t)}$ to encourage a greater change in state.

1.5.3 Hamiltonian Monte Carlo

We can reduce the random walk behaviour of standard Monte Carlo methods by incorporating gradient information from the probability density to drive proposals preferentially towards areas of higher probability. This can be achieved by inventing a fictional dynamical system in which "position" variables correspond to the states $\mathbf{x}$ of

interest, and auxiliary "momenta" $\mathbf{p}$ describe the rate of change of state variables. A random initial "flick" of the state imparts momentum which evolves as defined by the Hamiltonian dynamics, and after a finite number of discrete simulated timesteps, the new state is accepted according to the Metropolis rule. Because of the persistence of momentum from the initial flick, the state evolves in a more ordered manner than the random walk of standard methods.

In more detail, we write the probability from which we wish to sample as

$$p(\mathbf{x}) = \frac{1}{Z_p} \exp\left(-E(\mathbf{x})\right),$$

where $E(\mathbf{x})$ is interpreted as the "potential energy" of a state $\mathbf{x}$. The acceleration defined as the rate of change of momentum is due to the curvature of the probability surface and given by the negative gradient of the potential energy:

$$\frac{\mathrm{d}p_d}{\mathrm{d}\tau} = \frac{\partial E(\mathbf{x})}{\partial x_d}.$$

The momenta, defined as $p_d = \frac{\mathrm{d}x_d}{\mathrm{d}\tau}$, contribute "kinetic energy"

$$K(\mathbf{p}) = \frac{1}{2}\|\mathbf{p}\|^2,$$

yielding the *Hamiltonian* or total energy

$$H(\mathbf{x}, \mathbf{p}) = E(\mathbf{x}) + K(\mathbf{p}).$$

Since the Hamiltonian is separable, by generating samples from the joint probability whose energy is H, we can obtain samples from $p(\mathbf{x})$ simply by discarding the momentum variables.

Initially, a fresh sample is drawn for $\mathbf{p}$ from its Gaussian prior $\exp(-K(\mathbf{p}))/Z_K$, essentially a Gibbs sample which is always accepted. Next we propose a change in $\mathbf{x}$ by evaluating a sequence of discrete time approximations to the evolution of the dynamical system:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\tau} = \mathbf{p}; \qquad\qquad \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}\tau} = -\frac{\partial E(\mathbf{x})}{\partial \tau}.$$

Algorithm 1 Hamiltonian Monte Carlo

1: $\mathbf{g} = \nabla E(\mathbf{x}^{(0)})$
2: $E = E(\mathbf{x}^{(0)})$
3: **for** $t = 1$ to T **do**
4: $\quad \mathbf{p} = \mathcal{N}(\mathbf{p}\,;\,\mathbf{0}\,,\,\mathbf{I})$ {randomize momentum}
5: $\quad H = \frac{1}{2}\|\mathbf{p}\|^2 + E$
6: $\quad \mathbf{x}_\star = \mathbf{x};\ \mathbf{g}_\star = \mathbf{g}$
7: $\quad$ **for** $\tau = 1$ to τ_{max} **do** {simulate Hamiltonian dynamics to time τ_{max}}
8: $\quad\quad \mathbf{p} = \mathbf{p} - \epsilon\mathbf{g}_\star/2$
9: $\quad\quad \mathbf{x}_\star = \mathbf{x}_\star + \epsilon\mathbf{p}$
10: $\quad\quad \mathbf{g}_\star = \nabla E(\mathbf{x}_\star)$
11: $\quad\quad \mathbf{p} = \mathbf{p} - \epsilon\mathbf{g}_\star/2$
12: $\quad$ **end for**
13: $\quad E_\star = E(\mathbf{x}_\star)$
14: $\quad H_\star = \frac{1}{2}\|\mathbf{p}\|^2 + E_\star$ {evaluate Hamiltonian of new state}
15: $\quad$ accept with probability $\min(1, \exp\left(-(H_\star - H)\right))$
16: $\quad$ **if** accept **then**
17: $\quad\quad \mathbf{g} = \mathbf{g}_\star;\ \mathbf{x} = \mathbf{x}_\star;\ E = E_\star$
18: $\quad$ **end if**
19: **end for**

In a perfect simulation, i.e. by using sufficiently small step size ϵ, the Hamiltonian H remains constant and we would always accept the update. However, the discretization introduces errors which accumulate in the intermediate $\mathbf{x}$ and $\mathbf{p}$, and we use the acceptance criterion (1.19) to reject states which have a lower probability. In order to minimize these errors, it is common to employ "leapfrog" steps in which updates to $\mathbf{p}$ and $\mathbf{x}$ are interleaved. After $\tau_{\text{max}}/\epsilon$ steps we will have evolved the system to time τ_{max}. The relevant code appears on lines 8–13 in algorithm 1, which summarizes the entire process. Further details on Hamiltonian (elsewhere "hybrid") Monte Carlo can be found in Neal (1993), Andrieu et al. (2003), and MacKay (2003).

1.6 Structure of the thesis

This introductory chapter has presented some important results from the core theory of probabilistic machine learning. More advanced material directly relevant to the research has been collated in the subsequent chapters.

- **Chapter 2: Sparse Gaussian process classification**. This chapter explores in detail how the so-called FITC model for Gaussian processes, which constructs a

low-rank-plus-diagonal approximation to the covariance, can be extended from the case of Gaussian noise (a model which appeared in Snelson and Ghahramani (2006a) as the "pseudo-input Gaussian process") to arbitrary noise models, using expectation propagation to drive the inference. We focus on binary classification although our algorithm is widely applicable. Some of this material appeared in Naish-Guzman and Holden (2008a).

- **Chapter 3: Robust Gaussian process regression**. In this chapter, we describe a new model for robust regression which uses a secondary gating process indirectly to model the noise on the data. It is described how inference by expectation propagation is analytic, providing considerable speed advantage when compared with the Monte Carlo methods required by more complex GP mixtures. This model, the "twinned Gaussian process", appeared first in Naish-Guzman and Holden (2008b).

- **Chapter 4: Extending the twinned Gaussian process**. We further develop ideas from chapter 3, presenting a family of related GP mixture models in which all inference can be conducted by EP. Our examples include noise models for classification, regression and more general mixture modelling.

- **Chapter 5: Conclusions**, in which appear our concluding remarks.

- **Appendices A–C**. We have elected to present the longer derivations and proofs in appendices, allowing the elaboration of our ideas to flow more freely. Appendix A summarizes some mathematical preliminaries involving exponential families, Gaussian distributions and matrix algebra; appendix B collates the material from chapter 2; and appendix C does likewise for chapter 3.

- **Appendix D**. This appendix describes a method of constructing kernels from a basis class of threshold functions. By placing a novel prior on the class of linear halfspaces, we are able to integrate over their evaluation at arbitrary inputs to give a new non-stationary kernel function. Much of the material first appeared in Naish-Guzman et al. (2005); we have placed it in the appendix due to its more distant relationship to the main themes of the thesis.

CHAPTER 2

Sparse Gaussian process classification

PRINCIPAL AMONG THE challenges facing a practical application of Gaussian processes is the computational overhead of learning; we have seen how this procedure scales with N^3 for N inputs, and how prediction takes $\mathcal{O}(N^2)$. In recent years, there has been great interest in attempting to summarise the full GP using only a fraction of points $M \ll N$ known as the *inducing inputs* or *active set*, yielding a sparse predictor whose training and test times scale typically like $\mathcal{O}(NM^2)$ and $\mathcal{O}(M^2)$ respectively. Key to the ability of such methods is how this small set is chosen, and what probabilistic relationship is imposed between it, the remainder of the training data, and fresh test inputs. Beyond the realm of Gaussian noise, there is the additional consideration of how to handle an intractable posterior.

We identify in section 2.1 some recent approaches to sparsification, devoting most of our attention to the tractable case since more general likelihoods lend little insight and can obscure the fundamental concepts. We then examine in greater detail the model of Snelson and Ghahramani (2006a), which in section 2.2 is generalized to non-Gaussian likelihood distributions. The new model is applied to binary classification in section 2.3, followed in section 2.4 by a discussion. In section 2.5 we conclude with an evaluation of an extension intended for high-dimensional inputs, which reduces the number of hyperparameters to optimize by projecting data into a subspace.

2.1 Existing methods

The problem of speeding up training and prediction for GP models has been attacked from several directions, leading to a profusion of closely-related ideas presented by researchers with diverse backgrounds. Despite their assorted histories, it was illustrated by Quiñonero-Candela and Rasmussen (2005) that some of these ideas are very closely related. We adopt the unifying framework of their paper later in this chapter, but initially present the considerations required in the simplest method of all: it serves as a useful benchmark, and introduces concerns which apply also to many of the more complicated models.

2.1.1 Subset of data

If we ignore all but $M < N$ training inputs, we restrict time complexity to $\mathcal{O}(M^3)$ since inference is based on only M observations. When the noise model is Gaussian, predictions are made

$$p(\mathbf{f}_\star|\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}, \mathbf{y}_M) = \mathcal{N}\big(\mathbf{f}_\star \,;\, \mathbf{K}_{\star\bar{\mathbf{f}}}(\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}} + \sigma^2\mathbf{I})^{-1}\mathbf{y}_M \,,\, \mathbf{K}_{\star\star} - \mathbf{K}_{\star\bar{\mathbf{f}}}(\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\bar{\mathbf{f}}\star}\big)\,,$$

requiring the inversion of an $M \times M$ matrix only. In this notation, $\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}$ is the evaluation of the covariance function at only the chosen subset, in later contexts identified as the "active set"; $\mathbf{K}_{\bar{\mathbf{f}}\star}$ is the covariance matrix of correlations between the test data and elements of this subset; $\mathbf{y}_M$ are the observations restricted to the active set. This simplification allows a considerable speed-up, and if the data exhibit redundancy it need not compromise performance, but in general we run the risk of losing important information from the training set. Although simplistic, the subset of data (SD) method shares common considerations with its more advanced cousins below; a crucial question is: how should the active set be chosen?

For any measure of goodness of fit, finding the optimal solution is a combinatorial problem: there are $\binom{N}{M}$ subsets of size M. To avoid the exponential complexity that results, we could draw the active set entirely at random but this is unlikely to be effective in general. We may alternatively run a clustering algorithm and use directly the centres obtained for the inducing inputs. A more principled approach is to build the set greedily, including into it at each iteration that point which (in some sense) best im-

proves the approximation.[1] Several metrics for "improvement" have been suggested, some of which are extremely quick to evaluate and allow an optimal choice with respect to all the remaining data; others are rather expensive, and allow evaluation of only a fixed-size subset to retain a linear scaling with N.

Conceptually, we may imagine the most informative point should be that which causes greatest decrease in uncertainty of the current approximation, as the broad prior collapses upon the sharper peak of the posterior. The informative vector machine (IVM), introduced by Lawrence et al. (2003), is an SD method that implements this idea with a very efficient $\mathcal{O}(1)$ measure based on either the differential entropy $H[p(f_n)] - H[p(f_n|y_n)]$ or the information gain $\mathsf{KL}\left(p(f_n|y_n)\|p(f_n)\right)$ of including a point into the active set. From the marginal distribution after an inclusion, $p(f_n|y_n) \propto p(y_n|f_n)p(f_n)$, we find that in the case of Gaussian noise σ^2 on observations and a current marginal distribution $p(f_n) = \mathcal{N}(f_n\,;\,\mu_n\,,\,\sigma_n^2)$, the variance of the posterior is $(\sigma^{-2} + \sigma_n^{-2})^{-1}$. Using the fact that the entropy of a Gaussian with variance σ^2 is $\frac{1}{2}\log(2\pi e\sigma^2)$, the differential entropy is $\frac{1}{2}\log(1 + \sigma_n^2/\sigma^2)$, a monotonically increasing function of σ_n^2. The inclusion rule is simple: pick the point with greatest marginal variance.

The process of inclusion can be handled efficiently if we adopt an appropriate representation for the approximate posterior. Expectation propagation (EP) provides a useful foundation for the IVM, since by clamping most of the site parameters to zero the associated basis functions are automatically pruned from the model. Careful consideration of the matrix algebra allows us never to represent explicitly the full covariance, and only parameters of sites actively involved in the approximation are stored. Efficiency is achieved by observing that an inclusion grows the matrix $\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}$ by a single row and column, possible in $\mathcal{O}(MN)$ by working with the partitioned matrices; details appear in Seeger (2003, sec. 4.4.1 and app. C.3.1). The further advantage of using an EP scheme is that we implicitly prescribe a method for site inclusions, namely moment matching, which can be applied to any factorizing likelihood (possibly requiring one-dimensional quadrature). As such, the IVM constitutes a fairly general class of models, and in particular can be extended easily to binary classification. However, as with all SD methods, after sufficient data have been accumulated into its active set, predictions are made using the inducing inputs alone. In this respect, SD is a compres-

[1] There are parallels here with the idea of *active learning* (MacKay, 1992b), in which the learner can pose questions about the distribution at particular inputs, and the challenge is to ask those whose answers will be most informative.

sion scheme: other points are influential only in the weak sense of having guided the manner in which the model was grown.

There remains the crucial issue of fitting hyperparameters $\boldsymbol{\theta}$ in SD methods. If we choose the inducing inputs randomly then the SD approximation to the evidence

$$p(\mathbf{y}|\mathbf{X}) \approx \mathcal{N}\big(\mathbf{y}\,;\,\mathbf{0}\,,\,\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}} + \sigma^2\mathbf{I}\big)$$

can be optimized, with the proviso that with insufficient points included the approximation will be rather poor. If we use a greedy algorithm to grow the set incrementally, more care is needed because of interference between the choice of inclusion and the optimal $\boldsymbol{\theta}$: variations in the hyperparameters will affect which point should next be included; conversely, the growing active set may change at each iteration the optimal setting for $\boldsymbol{\theta}$. Seeger et al. (2006) describe how the two sets of updates can be interleaved in the IVM, but acknowledge difficulties in convergence. A subtle problem arises in particular for the EP approximation to the marginal likelihood used in the IVM, since after inclusion the sites are not further refined, so the preconditions for the derivation of section 1.3.2 do not apply.

2.1.2 Reduced rank methods

To improve on SD we need somehow to account for the data outside the active set. A recurring expression in the literature is the so-called *Nyström approximation*

$$\mathbf{K}_{\mathbf{ff}} \approx \mathbf{Q}_{\mathbf{ff}} \doteq \mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}, \tag{2.1}$$

which is an approximation of maximum rank M to the full covariance, given in terms of $\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}$ (the covariance of elements in the active set) and $\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}$ (correlations between the active set and the remaining data). In general, we will write $\mathbf{Q}_{\mathbf{ab}} \doteq \mathbf{K}_{\mathbf{a}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{b}}$. This expression can be used to advantage directly as an approximation to $\mathbf{K}_{\mathbf{ff}}$ because inversions can be speeded up with the matrix inversion lemma (A.2):

$$(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1} \approx (\mathbf{Q}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1} = \sigma^{-2}\mathbf{I} - \sigma^{-2}\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}(\sigma^2\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}} + \mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}})^{-1}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}.$$

The form (2.1) has been derived via an eigendecomposition of $\mathbf{K}$ in which only M eigenvectors are retained (Williams and Seeger, 2001; Rasmussen and Williams, 2006, sec. 8.1). Smola and Schölkopf (2000) consider instead an optimization problem in

which the kernel at each of the training points is approximated by a linear combination of the kernels $k(\mathbf{x}^{(m)}, \cdot)$ associated with the active set. In this setting, (2.1) arises from a particular error criterion that minimizes the deviation between the true kernel functions and the induced approximations; the active set is grown greedily in an effort to minimize the criterion.

Several authors (Lawrence et al., 2003; Seeger et al., 2003; Seeger, 2003) have regarded the task of finding a sparse model as making an approximation to the likelihood. Cast as an optimization involving the divergence $\mathrm{KL}(q(\mathbf{f}|\mathbf{y})\|p(\mathbf{f}|\mathbf{y}))$ (where $q(\mathbf{f}|\mathbf{y}) \propto p(\mathbf{f})\tilde{q}(\mathbf{y}|\bar{\mathbf{f}})$, and the likelihood approximation $\tilde{q}$ is restricted to a function of $\bar{\mathbf{f}}$), Seeger derives a model he calls *projected latent variables* (PLV). The "projection" occurs through the incorporation by $\tilde{q}$ of elements outside the active set: rather than ignore them as in SD, their latent $\mathbf{f}$ are modelled at the predictive mean that the SD method would provide (in the Gaussian case with the addition of noise σ^2), to provide an improved approximation to the marginal likelihood. Observe two properties of this measure: first, the divergence is the "wrong way around", in the sense discussed in section 1.3. Second, it is with respect to the posterior distribution, i.e. we have included the noise model in the heart of the approximation, which by this interpretation obscures a fully generic interface.[2]

For projected process models there arises the issue of how to select the active set. Smola and Bartlett (2001) suggest a greedy approach which takes as criterion the quadratic term from the log marginal likelihood $\mathbf{y}^T \left(\mathbf{Q}_{\mathbf{ff}} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y}$. As well as being prohibitively expensive to evaluate, allowing only a small number of random probes to avoid quadratic scaling with N, Quiñonero-Candela (2004, sec. 3.3.5) shows how the criterion lacks the evidential regularization required to avoid overfitting. Seeger et al. (2003) suggest an alternative greedy strategy, very similar to that employed for the IVM, which uses a cheap approximation to the information gain (the full version being too expensive to compute). Csató and Opper (2002) consider instead an online setting in which training data are presented individually. There, the decision of whether to include a fresh input is made by measuring its "novelty" (defined as the predictive variance according to the sparse model) and comparing it to a simple threshold. If the active set grows beyond some limit M, it is also described how less informative points can be removed to stay within memory bounds (Csató and Opper, 2002, sec. 3.3).

[2]However, we shall see an interpretation which divorces the likelihood from the sparsifying approximation in section 2.1.3.

2.1.3 Sparse methods as prior approximations

We turn now to the framework of Quiñonero-Candela and Rasmussen (2005), which allows us more easily to appreciate the restrictions imposed by PLV and compare it to similar models. This framework consists of simplified GP models, all understood in terms of different approximations to the *prior*. On reflection, this seems more natural than Seeger's interpretation of sparse models as posterior approximations, since the factorizing likelihood does not introduce any computational burden and in isolation could be represented exactly. Training a GP is difficult because of the densely connected prior covariance: only by making (implicitly or explicitly) a factorizing assumption can the cost of its manipulation be reduced. The new perspective provides the further benefit that any likelihood can be "plugged in" to such approximations without changing the theoretical basis, leading to more modular algorithms.

Suppose our GP includes a set $\bar{\mathbf{X}} \subset \mathcal{X}$ of inducing inputs, possibly but not necessarily a subset of the training data, and observe that by the consistency of GPs, the joint prior over latent values corresponding to training and test data $(\mathbf{f}, \mathbf{f}_\star)$ can be recovered by marginalizing over $\bar{\mathbf{f}}$, the function values at $\bar{\mathbf{X}}$:

$$p(\mathbf{f}, \mathbf{f}_\star) = \int p(\mathbf{f}, \mathbf{f}_\star | \bar{\mathbf{f}}) \mathcal{N}\big(\bar{\mathbf{f}}\,;\, \mathbf{0}\,,\, \mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}\big)\, \mathrm{d}\bar{\mathbf{f}}.$$

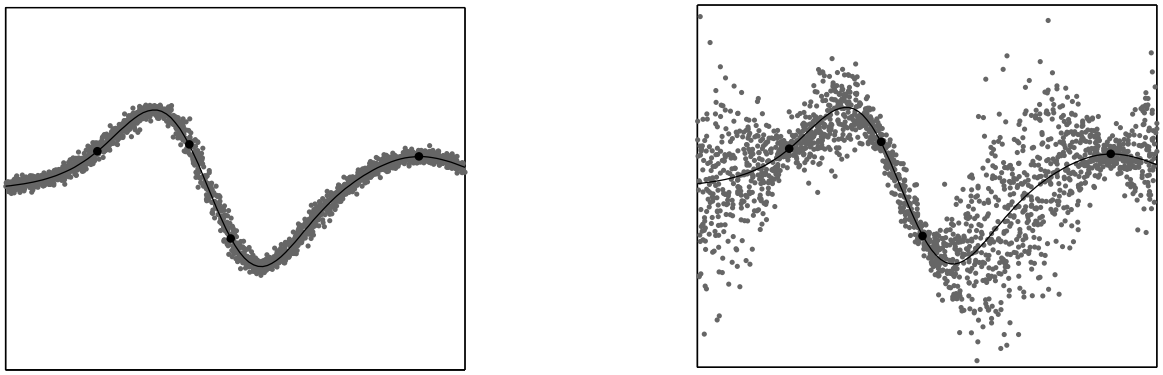
Quiñonero-Candela and Rasmussen (2005) reveal the nature of many sparse approximations to be a decomposition of the conditional prior $p(\mathbf{f}, \mathbf{f}_\star | \bar{\mathbf{f}})$ into separate training and test conditionals, yielding an approximation to the joint prior over $(\mathbf{f}, \mathbf{f}_\star)$

$$q(\mathbf{f}, \mathbf{f}_\star) = \int p(\mathbf{f}|\bar{\mathbf{f}}) p(\mathbf{f}_\star|\bar{\mathbf{f}}) \mathcal{N}\big(\bar{\mathbf{f}}\,;\, \mathbf{0}\,,\, \mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}\big)\, \mathrm{d}\bar{\mathbf{f}}. \tag{2.2}$$

This restriction severs the direct connection between training and test cases, forcing all communication through the bottleneck of the inducing inputs. A variety of schemes can then be recovered by making further approximations to each of the two conditional distributions; for example, PLV uses a deterministic approximation to the training conditional (so-called DTC)

$$q_{\mathrm{DTC}}(\mathbf{f}|\bar{\mathbf{f}}) = \mathcal{N}\big(\mathbf{f}\,;\, \mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\bar{\mathbf{f}}\,,\, \mathbf{0}\big)\,,$$

but retains the exact test conditional $q(\mathbf{f}_\star|\bar{\mathbf{f}}) = p(\mathbf{f}_\star|\bar{\mathbf{f}})$. Consider the generative model: we sample $\bar{\mathbf{f}}$ from the true prior, but fix $\mathbf{f}$ at the mean of the predictive process, adding



(a) DTC (b) FITC

Figure 2.1: Fantasy data drawn from two sparse generative models, parameterized by the inducing inputs denoted by black dots. The mean of $\mathbf{f}$ appears as a black line.

i.i.d. signal noise to obtain observations $\mathbf{y}$. Viewed in this way, it is clear that the method inherently underestimates variance away from the inducing inputs: where the generative variance should grow to that of the prior, it is instead fixed to the variance on observations (see fig. 2.1a). In a low-noise regime this is especially problematic for estimates of the marginal likelihood, since the prior cannot explain significant deviations from zero in regions away from the active set. The predictive distribution can also be regarded as faulty: when the inducing inputs are a subset of the data (as was the case until the work of Snelson and Ghahramani (2006a), see below), the mean prediction fails to interpolate correctly, noticeable primarily in environments with minimal noise (see Snelson, 2007, sec. 2.3.8).

An even simpler model, the *subset of regressors*, was introduced by Silverman (1985), and reappeared in Wahba et al. (1999) and Smola and Bartlett (2001). It shares the training conditional of PLV, but uses additionally a deterministic approximation for the test conditional

$$q_{\text{DIC}}(\mathbf{f}_\star|\bar{\mathbf{f}}) = \mathcal{N}\big(\mathbf{f}_\star\,;\,\mathbf{K}_{\star\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\bar{\mathbf{f}}\,,\,\mathbf{0}\big)\,.$$

The model becomes equivalent to exact inference in a GP with the degenerate kernel $k(\mathbf{x},\mathbf{z}) = Q(\mathbf{x},\mathbf{z}) \doteq \mathbf{k}(\mathbf{x},\bar{\mathbf{X}})\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\mathbf{k}(\bar{\mathbf{X}},\mathbf{z})$. There are some fundamental problems associated with degeneracy, identified several times in the literature (for example, in Quiñonero-Candela and Rasmussen, 2003), and which we discuss in section 2.1.4.

Snelson and Ghahramani (2006a) propose a more elaborate model in which the shortcomings of PLV are addressed, and which they call the "sparse pseudo-input GP" (SPGP). It consists of two distinct innovations: first, the approximation to the covariance of the conditional distribution $p(\mathbf{f}|\bar{\mathbf{f}})$ is enriched; second, they allow the active set to be located freely in the input space rather than restricted to a subset of the training data. The generative process is similar to DTC, but the $\mathbf{f}$ appear now as independent samples drawn from the true predictive process $p(f_n|\mathbf{x}_n, \bar{\mathbf{f}})$, which amounts to a full independence assumption on the training conditional; as such, Quiñonero-Candela and Rasmussen (2005) use the acronym FITC.[3]

$$q_{\text{FITC}}(\mathbf{f}|\bar{\mathbf{f}}) = \mathcal{N}\big(\mathbf{f}\,;\, \mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\bar{\mathbf{f}}\,,\, \mathsf{diag}\left(\mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}\right)\big)\,;$$

the correlations of the full GP have been lost, but the marginal variances are exact: see fig. 2.1b. In terms of divergence, Snelson (2007) observes that the FITC solution is obtained by minimizing $\mathsf{KL}(p(\mathbf{f},\bar{\mathbf{f}})\|q(\mathbf{f},\bar{\mathbf{f}}))$ subject to the constraint that the $q(\mathbf{f}|\bar{\mathbf{f}})$ factorize (in which case $q(f_n|\bar{\mathbf{f}}) = p(f_n|\bar{\mathbf{f}})$). Observe that this divergence measure seems more appropriate than that employed in PLV, and that the likelihood has not yet entered the model.

In his thesis, Snelson establishes that the work of Csató and Opper (2002) was in fact first to introduce the FITC approximation, something obscured for many years by its very different presentation and motivation as an online method. The model is initialized at the prior and uses a moment matching scheme for non-Gaussian likelihoods (which is exact in the tractable case). By processing the data sequentially, it also enforces the factorizing assumption central to FITC. However, unlike the SPGP and its generalization here, the FITC evidence approximation is not used to set hyperparameters, nor of course more generally to optimize the placement of points in the active set, which in Csató and Opper (2002) are training inputs deemed sufficiently "novel" to be accumulated into $\bar{\mathbf{X}}$.

By using an approximation that more faithfully represents the true covariance $\mathbf{K}_{\mathbf{ff}}$ we can develop even more accurate models. One approach that assures us of a positive

[3] In the original paper, Snelson and Ghahramani place the pseudo-inputs randomly and learn their locations by non-linear optimization of the marginal likelihood. Here, the term FITC is used to refer to the derived covariance structure only, and SPGP to imply the additional use of gradient information to locate the active set. As emphasized in Quiñonero-Candela and Rasmussen (2005), the FITC approximation is applicable regardless of how the inducing inputs are obtained, and other schemes for their initialization could equally well be married with the algorithm.

definite result is to use block elements along the diagonal. In this case, we recover PITC or the "partially independent training conditional" assumption, which appears in a different guise at the core of the Bayesian committee machine (BCM) introduced by Tresp (2000). A second innovation of the BCM was its transductive nature: the active set is simply fixed at the test inputs. However, analogously to the separation of pseudo-input optimization and FITC approximation, the new perspective on sparse GP models makes it clear that the concepts of blocking and transduction can be treated separately—indeed, it is much less clear that the transductive element is even helpful; for example it delays until testing much of the "learning" that a GP requires, causing complexity of prediction to scale with N. Snelson and Ghahramani (2007) extend the blocking structure to the test inputs also, effectively partitioning the input into a collection of loosely coupled GPs; by analogy, the acronym PIC is used.

2.1.4 Relevance vector machines

The relevance vector machine (RVM) of Tipping (2001) is a general attempt to encourage sparsity in linear models. By attaching an individual weight to each component and placing on them a diagonal Gaussian prior, in the maximum likelihood limit of optimizing all these extra parameters it is often found that many are driven to zero, pruning the corresponding functions from the model.[4] There is something curious about this process, which in certain circles sparks lively debate as to whether it even qualifies as "Bayesian": the model is likely not one we believe in, and all the additional weights should strictly be marginalized and not optimized (which of course would nullify the computational advantage).

However, there are issues beyond the conceptual: the method can be understood as a special form of degenerate GP (see Rasmussen and Williams, 2006, sec. 6.6), and therefore inherits some undesirable features. Although it tends to fit the mean process accurately and with remarkable sparsity, its variance predictions are nonsensical, shrinking to zero away from the data for radial basis functions (where all of the kernels predict near-zero signal), and in general making predictions of limited flexibility. This curious property is explored in Quiñonero-Candela (2004, sec. 2.5.1) and Rasmussen and Quiñonero-Candela (2005), where *augmentation* is used to "heal" its

[4] The original paper was inefficient in its optimization, starting with a full model and removing components; subsequent work by Tipping and Faul (2003) presented a superior algorithm, similar to *sequential minimal optimization* used to train support vector machines, which starts with an empty model and adds components one by one.

variance estimates: during testing, an extra basis function is placed at the test point to restore predictive uncertainty in regions distant from the training set. This provides a useful illustration of the failure of the RVM, but test costs scale $\mathcal{O}(NM)$ making it an impractical sparse method. An ingenious alternative appears in Quiñonero-Candela et al. (2007), where the introduction of an additional white noise process, and subsequent renormalization across the bases to constant prior variance, has the effect of decorrelating samples away from the basis functions, and restoring the predictive uncertainty—all without affecting the computational demands of inference. However, the illustrations in their technical report reveal certain glitches in the variance predictions due to the unusual construction, and it would be interesting to see how these behave beyond the unidimensional case.

When we move to the classification domain, it is difficult to assess the importance of accurately predicting the variance of the latent function, since the mean already contains implicit uncertainty about the class assignment. With the RVM for example, the decay to zero in the latent signal corresponds to what probably constitutes a fairly safe assumption, namely that away from the data, either class could occur with equal probability. In general too, the variance on the latent mean amounts to a secondary source of uncertainty which for predictions is essentially assimilated directly into the probabilistic estimate, such that large values of $\sigma_\star^2$ pull the predictions $p(y_\star|\mathbf{x}_\star)$ towards 0.5. However, we are reluctant to advocate the RVM for classification, since even near the data these estimates behave counterintuitively, typically *growing* to their maximum value precisely at the basis functions. It is in these regions, where data are expected to occur, that we should hope for greatest accuracy and, where applicable, near-certainty.

2.1.5 Support vector machines

Finally, we mention the well-established workhorse of the kernel methods community, the support vector machine (SVM) (Vapnik, 1995). Although born of frequentist arguments from statistical learning theory, the SVM bears a strong resemblance to GP classification via its use of a kernel function as a measure of similarity.[5] Driven by non-Bayesian, essentially geometric considerations, the SVM is the hyperplane that provides greatest separation between points of opposite class. In the separable case, it

[5]Indeed, Sollich (2002) derives an unusual Bayesian interpretation, despite the fact that the "hinge" loss used in SVMs does not correspond to any negative log likelihood.

arises from the solution of the quadratic programming (QP) problem

$$\begin{aligned}&\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2\\&\text{subject to } y_n(\mathbf{w}^T\mathbf{x}_n + w_0) \geq 1 \text{ for } n \in \{1, 2, \ldots, N\}.\end{aligned}$$

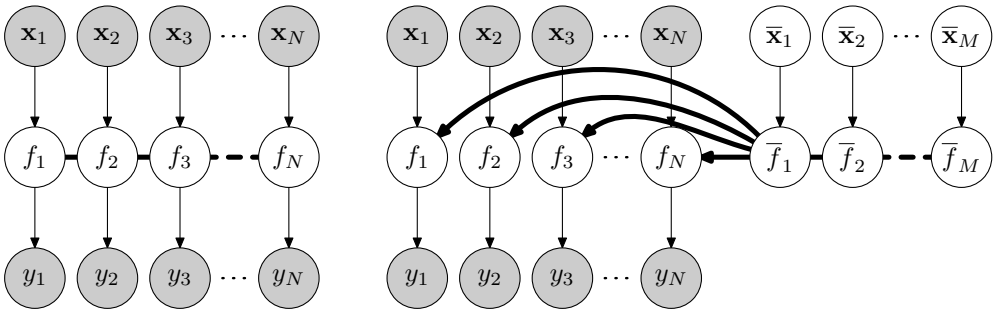
The form of the solution turns out to be $\mathbf{w} = \sum_{n=1}^{N} \lambda_n y_n \mathbf{x}_n$, where the $\boldsymbol{\lambda}$ are non-negative Lagrange multipliers. Although sparsity is not by design an objective of the SVM, this solution is found in general to yield a decision function involving comparatively few kernel evaluations because λ_n is positive only for data points $\mathbf{x}_n$ closest to the separating plane. Since the QP and predictions at fresh inputs involve only inner products, the "kernel trick" allows us to perform non-linear classification of data by lifting them into potentially infinite-dimensional feature spaces in which the vector $\mathbf{w}$ need never be calculated explicitly.

The inseparable case is solved by the soft margin SVM (Cortes and Vapnik, 1995),

$$\begin{aligned}&\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \text{ for fixed } C > 0\\&\text{subject to } y_n(\mathbf{w}^T\mathbf{x}_n + w_0) \geq 1 - \xi_n \text{ and } \xi_n \geq 0 \text{ for } n \in \{1, 2, \ldots, N\},\end{aligned}$$

which employs "slack variables" $\boldsymbol{\xi}$ in a computationally expedient addition that maintains an efficient training algorithm but allows a small number of misclassifications of the training set. However, sparsity and regularization are intimately linked by the single parameter C; in consequence, we find noisy data rarely admit very sparse solutions. Furthermore, the value this parameter should take must be estimated by cross-validation, since the model is not probabilistic, and has no concept of "evidence". (The standard SVM provides only a class label at test inputs, although Platt (1999) attempts to use the real-valued distance of training points from the hyperplane as a measure of probabilistic certainty by postfitting a sigmoid function across the boundary. Such ad hoc measures must leave the Bayesian feeling a little queasy.)

The prospect of parameterizing a decision boundary entirely in terms of data points that lie closest to it should also give us pause for thought. For example, if we sought to compress a catalogue of images of the digits 0 through 9, we would expect to make more efficient use of a library of prototypical images than one of illegible boundary cases. One expects the majority of data, drawn from the unknown $p(\mathbf{x})$, to fall away



(a) The full GP. (b) The FITC approximation.

Figure 2.2: Sparse GP approximations. Shaded variables are observed; bold lines indicate fully-connected variables. In the sparse setting, bold arrows indicate that each f_n is connected to every $\bar{f}_m$.

from this region of relative uncertainty, and we may imagine that the decision should be based more upon samples which are somehow "representative", located near the centre of class clusters, not on the particular boundary examples that happen to occur in the training set. Basis functions are typically local, and by placing them along the class boundary we are liable to preclude very sparse solutions since they exert considerable force in shaping the decision surface; only by using many can we avoid its undesirable warping. A smoother surface will tend to arise only from a parameterization involving fewer and likely more distant points. We return to this aspect of sparse solutions to classification problems in section 2.4.

2.2 The generalized FITC approximation

Following Snelson and Ghahramani (2006a), we place a GP prior over a set of M inducing inputs $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots, \bar{\mathbf{x}}_M\}$, from which is drawn a sample $\bar{\mathbf{f}} = \{\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_M\}$. At each of the N real data points $\mathbf{x}_n$, the latent function value f_n is drawn independently from the posterior GP obtained by conditioning on $\bar{\mathbf{f}}$, and the observations are generated i.i.d. from the likelihood $p(y_n|f_n)$. The graphical model is illustrated in Fig. 2.2b.

We saw above that this model can be understood as a particular approximation to the prior, and this perspective reveals an efficient algorithm for inference. Let

$$q(\mathbf{f}|\bar{\mathbf{f}}, \mathbf{X}) = \mathcal{N}\big(\mathbf{f}\,;\, \mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\bar{\mathbf{f}}\,,\, \mathsf{diag}\left(\mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}\right)\big)\,, \tag{2.3}$$

$$q(\mathbf{f}_\star|\bar{\mathbf{f}}, \mathbf{X}_\star) = \mathcal{N}\big(\mathbf{f}_\star\,;\, \mathbf{K}_{\star\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\bar{\mathbf{f}}\,,\, \mathsf{diag}\left(\mathbf{K}_{\star\star} - \mathbf{Q}_{\star\star}\right)\big)\,. \tag{2.4}$$

Predictions require the posterior distribution over inducing inputs $\bar{\mathbf{f}}$; this is most efficiently obtained via Bayes' rule after inferring the distribution over $\mathbf{f}$. (We note that one could also infer the posterior over $\bar{\mathbf{f}}$ directly, rather than marginalizing over the inducing inputs as here. Running EP in this setting, each site maintains a belief about the full $M \times M$ covariance, and we obtain a slower $\mathcal{O}(NM^3)$ algorithm. Furthermore, calculations to evaluate the derivatives of the log marginal likelihood with respect to inducing inputs $\bar{\mathbf{x}}_m$ are significantly complicated by their presence in both prior and likelihood.)

Using (2.3) and marginalizing over the exact prior on $\bar{\mathbf{f}}$ we obtain the prior on $\mathbf{f}$ imposed by the model,

$$\begin{aligned} q(\mathbf{f}|\mathbf{X}, \bar{\mathbf{X}}) &= \int \mathcal{N}\big(\mathbf{f}\,;\, \mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\bar{\mathbf{f}}\,,\, \mathsf{diag}\left(\mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}\right)\big)\, \mathcal{N}\big(\bar{\mathbf{f}}\,;\, \mathbf{0}\,,\, \mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}\big)\, \mathrm{d}\bar{\mathbf{f}} \\ &= \mathcal{N}(\mathbf{f}\,;\, \mathbf{0}\,,\, \mathbf{Q}_{\mathbf{ff}} + \mathsf{diag}\left(\mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}\right))\,, \end{aligned} \tag{2.5}$$

whose covariance consists of the sum of a low-rank term $\mathbf{Q}_{\mathbf{ff}}$ and a diagonal matrix. Given a Gaussian (approximation to the) posterior $p(\mathbf{f}|\mathbf{X}, \mathbf{y})$, the posterior over the pseudo-inputs

$$p(\bar{\mathbf{f}}|\mathbf{X}, \mathbf{y}, \bar{\mathbf{X}}) = \int p(\bar{\mathbf{f}}|\bar{\mathbf{X}}, \mathbf{f}) p(\mathbf{f}|\mathbf{X}, \mathbf{y}) \mathrm{d}\mathbf{f}$$

can also be written in a Gaussian form.

Snelson and Ghahramani (2006a) describe the tractable case, when the noise model is Gaussian. In this chapter, we explore an approach for handling non-Gaussian likelihoods, considering as an example probit noise for binary classification. This is not only a common problem, but our results in section 2.3 bear out the intuition that sparse methods should be well-suited: many data sets enjoy the property that large regions of the input space are predominantly of only one class, hence the latent signal may be assumed broadly constant in these regions. This is in contrast to the latent process in regression tasks, which must follow the higher-frequency, continuous behaviour of the observations. We see from the diagonal covariance in (2.3), and from the example in fig. 2.2b, that in no other way but an increased density of pseudo-inputs can correlations be introduced into $\mathbf{f}$. Classification problems are therefore uniquely interesting, being in this sense most amenable to sparse representations.

2.2.1 Inference

In the discussion below we omit the moment calculations for particular noise models; they can be transplanted essentially unchanged from the corresponding dense GP. The calculations required to implement the probit model, used extensively in our experiments, appear in appendix B. Instead, we describe only how the mean and covariance structure of the approximate posterior is preserved for a general likelihood. The covariance in the prior (2.5) is the sum of a diagonal component $\mathbf{D}$ and a rank-M term $\mathbf{PMP}^T$, where $\mathbf{P}_0 = \mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}$ and $\mathbf{M}_0 = \mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}$ (these zero subscripts refer to the initial values of the matrices $\mathbf{D}$, $\mathbf{P}$ and $\mathbf{M}$, although they are updated by EP during the course of the iterations). Since the observations y_n are generated i.i.d., we can expect this decomposition to persist in the posterior.

EP requires efficient operations for marginalization to obtain $p(f_n)$; for updating the posterior distribution after refining a site; and when applicable, for refreshing the posterior to avoid loss of numerical precision. Decomposing $\mathbf{M} = \mathbf{R}^T\mathbf{R}$ into its Cholesky factor,[6] we represent the posterior covariance $\mathbf{A}$ and mean $\mathbf{h}$ by

$$\mathbf{A} = \mathbf{D} + \mathbf{PR}^T\mathbf{RP}^T \qquad \text{and} \qquad \mathbf{h} = \boldsymbol{\zeta} + \mathbf{P}\boldsymbol{\gamma},$$

where $\mathbf{D}$ is diagonal, $\boldsymbol{\zeta}$ is $N \times 1$ and $\boldsymbol{\gamma}$ is $M \times 1$; these latter parameters are initialized $\boldsymbol{\zeta} = \mathbf{0}$ and $\boldsymbol{\gamma} = \mathbf{0}$. Writing $\mathbf{p}_n^T = \mathbf{P}_{(n,\cdot)}$ and $d_n = D_{nn}$, we obtain

$$A_{nn} = d_n + \|\mathbf{Rp}_n\| \qquad \text{in } \mathcal{O}(M^2); \qquad h_n = \nu_n + \mathbf{p}_n^T\boldsymbol{\gamma} \qquad \text{in } \mathcal{O}(M). \tag{2.6}$$

Now consider a change in the precision at site n by π_n. Define the vector $\mathbf{e}$ of length N such that $e_n = 1$ and all other elements are zero. The new covariance $\mathbf{A}_{\text{new}}$ is obtained by inverting the sum of the old precision matrix and the change in precision. If we let $\mathbf{E} = \mathbf{D}^{-1} + \pi_n\mathbf{ee}^T$, so that

$$\mathbf{E}^{-1} = \mathbf{D} - \frac{\pi_n d_n^2}{1 + \pi_n d_n}\mathbf{ee}^T \qquad \text{and} \qquad (\mathbf{DED})^{-1} = \mathbf{D}^{-1} - \frac{\pi_n}{1 + \pi_n d_n}\mathbf{ee}^T,$$

[6] Care must be taken that the factors share the correct orientation. When the environment (e.g. Matlab) offers only upper Cholesky factors $\mathbf{R}^T\mathbf{R}$, the initialization of $\mathbf{R}_0 = \mathsf{chol}\left(\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\right)$ can be achieved without computing the explicit inverse via the following matrix rotations:

$$\mathbf{R}_0 := \mathsf{rot180}\left(\mathsf{chol}\big(\mathsf{rot180}\left(\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}\right)\big)^T \setminus \mathbf{I}\right).$$

then from the matrix inversion lemma (A.2), and incorporating the update to site n,

$$\begin{aligned}
\mathbf{A}^{-1} &= \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{P}\mathbf{R}^T(\mathbf{R}\mathbf{P}^T\mathbf{D}^{-1}\mathbf{P}\mathbf{R}^T + \mathbf{I})^{-1}\mathbf{R}\mathbf{P}^T\mathbf{D}^{-1} \\
\implies \mathbf{A}_{\text{new}} &= \mathbf{E}^{-1} - \mathbf{E}^{-1}\mathbf{D}^{-1}\mathbf{P}\mathbf{R}^T\times \\
&\qquad \left(\mathbf{R}\mathbf{P}^T(\mathbf{D}\mathbf{E}\mathbf{D})^{-1}\mathbf{P}\mathbf{R}^T - \mathbf{I} - \mathbf{R}\mathbf{P}^T\mathbf{D}^{-1}\mathbf{P}\mathbf{R}^T\right)^{-1}\mathbf{R}\mathbf{P}^T\mathbf{D}^{-1}\mathbf{E}^{-1} \\
&= \mathbf{D}_{\text{new}} + \mathbf{P}_{\text{new}}\mathbf{R}_{\text{new}}^T\mathbf{R}_{\text{new}}\mathbf{P}_{\text{new}}^T,
\end{aligned}$$

where we expand the inversion in parentheses to obtain a rank-1 downdate to the Cholesky factor $\mathbf{R}$;[7] in summary

$$\mathbf{D}_{\text{new}} = \mathbf{D} - \frac{\pi_n d_n^2}{1+\pi_n d_n}\mathbf{e}\mathbf{e}^T \qquad \mathcal{O}(1) \text{ update}, \tag{2.7a}$$

$$\mathbf{P}_{\text{new}} = \mathbf{P} - \frac{\pi_n d_n}{1+\pi_n d_n}\mathbf{e}\mathbf{p}_n^T \qquad \mathcal{O}(M) \text{ update}, \tag{2.7b}$$

$$\mathbf{R}_{\text{new}} = \mathsf{chol}_{\downarrow}\left(\mathbf{R}^T\left(\mathbf{I} - \mathbf{R}\mathbf{p}_n\frac{\pi_n}{1+\pi_n A_{nn}}\mathbf{p}_n^T\mathbf{R}^T\right)\mathbf{R}\right) \qquad \mathcal{O}(M^2) \text{ update}. \tag{2.7c}$$

If the second site parameter, corresponding to precision times mean, is changed by b_n, then

$$\begin{aligned}
\mathbf{A}_{\text{new}}^{-1}\mathbf{h}_{\text{new}} &= \mathbf{A}^{-1}\mathbf{h} + b_n\mathbf{e} \\
\implies \mathbf{h}_{\text{new}} &= \mathbf{A}_{\text{new}}\left(\mathbf{A}_{\text{new}}^{-1} - \pi_n\mathbf{e}\mathbf{e}^T\right)\mathbf{h} + \mathbf{A}_{\text{new}}b_n\mathbf{e} \\
&= \boldsymbol{\zeta}_{\text{new}} + \mathbf{P}_{\text{new}}\boldsymbol{\gamma}_{\text{new}},
\end{aligned}$$

where

$$\boldsymbol{\zeta}_{\text{new}} = \boldsymbol{\zeta} + \frac{(b_n + \pi_n\nu_n)d_n}{1+\pi_n d_n}\mathbf{e} \qquad \mathcal{O}(1) \text{ update}; \tag{2.8a}$$

$$\boldsymbol{\gamma}_{\text{new}} = \boldsymbol{\gamma} + \frac{b_n - \pi_n h_n}{1+\pi_n d_n}\mathbf{R}_{\text{new}}^T\mathbf{R}_{\text{new}}\mathbf{p}_n \qquad \mathcal{O}(M^2) \text{ update}. \tag{2.8b}$$

It is necessary to refresh the covariance and mean every complete EP cycle to avoid loss of precision due to repeated low rank updates. This is achieved by incorporating

[7] If the factor $\frac{\pi_n}{1+\pi_n A_{nn}}$ is negative, we make a rank-1 *up*date, guaranteed to preserve the positive definite property. Note that on rare occasions, loss of precision can cause a downdate to result in a non-positive definite covariance matrix. If this occurs, we should abort the update and refresh the posterior from scratch. In any case, to improve conditioning, it is recommended to add a small multiple of the identity to the prior $\mathbf{M}_0$.

Algorithm 2 EP approximation to the FITC posterior

input: $\mathbf{X}, \mathbf{y}, \bar{\mathbf{X}}, \boldsymbol{\theta}$
initialize covariance: $\mathbf{D} := \mathsf{diag}\left(\mathbf{K_{ff}} - \mathbf{Q_{ff}}\right); \mathbf{P} := \mathbf{K_{f\bar{f}}}; \mathbf{R} := \mathsf{chol}(\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1})$
initialize mean: $\boldsymbol{\zeta} := \mathbf{0}; \boldsymbol{\gamma} := \mathbf{0}$
while L not converged **do**
  for all n **do**
    obtain marginal distribution using (2.6)
    obtain cavity distribution using (1.11)
    calculate Z_n, $\boldsymbol{\alpha}_n$, $\boldsymbol{\nu}_n$ using (1.12)
    determine new site parameters by (1.13)
    update posterior representation using (2.7) and (2.8)
  end for
  refresh posterior using (2.9) and (2.10)
  calculate the approximate log marginal likelihood L using (1.16)
end while
return: $\mathbf{D}; \mathbf{P}; \mathbf{R}; \boldsymbol{\zeta}; \boldsymbol{\gamma}; L$

all the site parameters directly into the prior.

$$\mathbf{D}_{\text{new}} = \left(\mathbf{I} + \mathbf{D}_0\boldsymbol{\Pi}\right)^{-1}\mathbf{D}_0 \qquad (\mathcal{O}(N)); \tag{2.9a}$$

$$\mathbf{P}_{\text{new}} = \left(\mathbf{I} + \mathbf{D}_0\boldsymbol{\Pi}\right)^{-1}\mathbf{P}_0 \qquad (\mathcal{O}(NM)); \tag{2.9b}$$

$$\mathbf{R}_{\text{new}} = \mathsf{rot180}\Big(\mathsf{chol}\big(\mathsf{rot180}\big(\mathbf{I} + \mathbf{R}_0\mathbf{P}_0^T\boldsymbol{\Pi}\left(\mathbf{I} + \mathbf{D}_0\boldsymbol{\Pi}\right)^{-1}\mathbf{P}_0\mathbf{R}_0^T\big)\big)^T\Big)\Big\backslash\mathbf{R}_0 \quad \big(\mathcal{O}(NM^2)\big), \tag{2.9c}$$

where $\mathbf{R}_{\text{new}}$ is obtained being careful to ensure the orientations of the factorizations are not mixed. Finally, the mean is refreshed using

$$\boldsymbol{\zeta}_{\text{new}} = \mathbf{D}_{\text{new}}\mathbf{b} \qquad \text{in } \mathcal{O}(N); \tag{2.10a}$$

$$\boldsymbol{\gamma}_{\text{new}} = \mathbf{R}_{\text{new}}^T\mathbf{R}_{\text{new}}\mathbf{P}_{\text{new}}^T\mathbf{b} \qquad \text{in } \mathcal{O}(NM), \tag{2.10b}$$

where we have assumed $\mathbf{h}_0 = \mathbf{0}$.

Reviewing algorithm 2, we see that EP costs are dominated by the $\mathcal{O}(M^2)$ Cholesky downdate at each site inclusion. After visiting each of the N sites, we are advised to perform a full refresh, which is $\mathcal{O}(NM^2)$, togther leading to asymptotic complexity of $\mathcal{O}(NM^2)$.

2.2.2 Model selection

Section 1.3 states how derivatives of the marginal likelihood can be estimated by EP. Unfortunately, we cannot use these terms directly in our classifier because the matrix products require $\mathcal{O}(N^2)$ space, and implemented naïvely require $\mathcal{O}(N^3)$ time. However, by taking advantage of the diagonal-plus-low-rank structure of the posterior covariance, these requirements can be reduced to $\mathcal{O}(NM)$ and $\mathcal{O}(NM^2)$ respectively. Further complications arise if we seek also to optimize the active set as in SPGP, since derivatives of matrices with respect to these vectors yield tensor results. Details of all these calculations are presented in appendix B, including a description of how the difficulty in this latter case can be overcome efficiently.

2.2.3 Predictions

As with most GP models, before predictions can be made the FITC approximation requires an initial series of precomputations once model selection is complete. In our model these calculations cost $\mathcal{O}(NM^2)$, but having stored the matrix results, all future predictions can be made in $\mathcal{O}(M^2)$, or $\mathcal{O}(M)$ if we are not interested in the variance. Hence we describe the predictive cost as $\mathcal{O}(M^2)$, treating as asymptotically unimportant in the limit of a large number of test points the cost associated with the precomputations; alternatively, they can be considered intrinsic to training.

First we marginalize out $\bar{\mathbf{f}}$ from (2.4). Initially, Bayes' theorem is used to find the posterior distribution over $\bar{\mathbf{f}}$ from the inferred posterior over $\mathbf{f}$:

$$p(\bar{\mathbf{f}}|\mathbf{f}) \propto p(\mathbf{f}|\bar{\mathbf{f}})p(\bar{\mathbf{f}}) = \mathcal{N}(\bar{\mathbf{f}} \,|\, \mathbf{R}_0^{-1}\mathbf{c}, \mathbf{R}_0^{-1}\mathbf{C}\mathbf{R}_0^{-T}),$$
$$\text{where} \quad \mathbf{c} = \mathbf{C}\mathbf{R}_0\mathbf{P}_0^T\mathbf{D}_0^{-1}\mathbf{f} \quad \text{and} \quad \mathbf{C}^{-1} = \mathbf{I} + \mathbf{R}_0\mathbf{P}_0^T\mathbf{D}_0^{-1}\mathbf{P}_0\mathbf{R}_0^T.$$

Let our posterior approximation be $q(\mathbf{f}|\mathbf{y}) = \mathcal{N}(\mathbf{f}\,;\,\mathbf{h}\,,\,\mathbf{A})$. Hence

$$p(\bar{\mathbf{f}}|\mathbf{y}) \approx \int p(\bar{\mathbf{f}}|\mathbf{f})q(\mathbf{f}|\mathbf{y})\mathrm{d}\mathbf{f} = \mathcal{N}(\bar{\mathbf{f}} \,|\, \mathbf{R}_0^{-1}\boldsymbol{\mu}, \mathbf{R}_0^{-1}\boldsymbol{\Sigma}\mathbf{R}_0^{-T}),$$
$$\text{where} \quad \boldsymbol{\mu} = \mathbf{C}\mathbf{R}_0\mathbf{P}_0^T\mathbf{D}_0^{-1}\mathbf{h} \quad \text{and} \quad \boldsymbol{\Sigma} = \mathbf{C} + \mathbf{C}\mathbf{R}_0\mathbf{P}_0^T\mathbf{D}_0^{-1}\mathbf{A}\mathbf{D}_0^{-1}\mathbf{P}_0\mathbf{R}_0^T\mathbf{C}.$$

Obtaining these terms is $\mathcal{O}(NM^2)$ if we take advantage of the structure of $\mathbf{A}$ as diagonal $\mathbf{D}$ plus low-rank $\mathbf{P}\mathbf{R}^T\mathbf{R}\mathbf{P}^T$.

Let the Cholesky factorization

$$\mathbf{C}^{-1} = \mathbf{V}\mathbf{V}^T,$$

and define

$$\mathbf{U} = \mathbf{V} \backslash \mathbf{R}_0 \mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}} \mathbf{D}_0^{-1},$$

obtained in $\mathcal{O}(NM^2)$, so that

$$\boldsymbol{\mu} = \mathbf{V}^T \backslash \mathbf{U}\mathbf{h}; \qquad \boldsymbol{\Sigma} = \mathbf{V}^T \backslash \big(\mathbf{I} + \underbrace{\mathbf{U}\mathbf{D}\mathbf{U}^T}_{\mathcal{O}(NM^2)} + \underbrace{\mathbf{U}\mathbf{P}\mathbf{R}^T\mathbf{R}\mathbf{P}^T\mathbf{U}^T}_{\mathcal{O}(NM^2)} \big) / \mathbf{V}.$$

The prediction of $f_\star$ at test point $\mathbf{x}_\star$ is

$$p(f_\star | \mathbf{x}_\star, \mathbf{y}) = \int p(f_\star | \bar{\mathbf{f}}) p(\bar{\mathbf{f}} | \mathbf{y}) \mathrm{d}\bar{\mathbf{f}} = \mathcal{N}(f_\star \,|\, \mu_\star, \sigma_\star^2),$$

where, after precomputations,

$$\begin{aligned} \mu_\star &= \mathbf{k}_{\star\bar{\mathbf{f}}} \mathbf{R}_0^T \boldsymbol{\mu} && \text{is } \mathcal{O}(M), \\ \sigma_\star^2 &= k_{\star\star} + \mathbf{k}_{\star\bar{\mathbf{f}}} \mathbf{R}_0^T \left(\boldsymbol{\Sigma} - \mathbf{I} \right) \mathbf{R}_0 \mathbf{k}_{\bar{\mathbf{f}}\star} && \text{is } \mathcal{O}(M^2). \end{aligned}$$

We will often be interested in the distribution of $y_\star$, which is dependent on the noise model. In the classification domain,

$$p(y_\star | \mathbf{x}_\star, \mathbf{y}) = \int p(y_\star | f_\star) p(f_\star | \mathbf{x}_\star, \mathbf{y}) \mathrm{d}f_\star = \sigma \left(\frac{y_\star (\mu_\star + b)}{\sqrt{1 + \sigma_\star^2}} \right).$$

2.2.4 Implementation

In practice, it must be established how to initialize the pseudo-inputs $\bar{\mathbf{X}}$. A variety of methods suggest themselves; we could place them on a random subset of the training data; we could perform a preliminary K-means clustering, perhaps once for each class, and initialize the $\bar{\mathbf{X}}$ at the cluster centres; we can also grow the set greedily, using one of the metrics from section 2.1; we might even use the $\bar{\mathbf{X}}$ in a "transductive" manner, placing them at the test points as in the Bayesian committee machine.

In our implementation we placed the inputs randomly on real data, and used gradient information to optimize their placement. The optimization of model parameters was performed with a conjugate gradients minimizer, but we rescaled the log marginal likelihood and its derivatives in order that the standard settings could be used effectively, dividing by N.

2.3 Experiments

We conducted tests on a variety of data, including two small sets from Ripley (1996)[8] and the benchmark suite of Rätsch.[9] The dimensionality of these classification problems ranges from two to sixty, and the size of the training sets is of the order of 400 to 1000. Results are presented in table 2.1. We include both the error rate and the predictive certainty, which for *crabs* and the Rätsch sets are averaged over ten folds of the data; for the *synth* problem, Ripley has already divided the data into training and test partitions.

Comparisons are made with the full GP classifier trained by EP,[10] and the SVM,[11] a discriminative model in practice found to yield relatively sparse solutions; we consider also the IVM, a popular framework for building sparse linear models.[12] In all cases, we employed the isotropic squared exponential kernel (1.3), avoiding here the anisotropic version primarily to allow comparison with the SVM: lacking a probabilistic foundation, its kernel parameters and regularization constant must be set by cross-validation. For the GP models, we fit hyperparameters by gradient ascent on the estimated marginal likelihood (limiting the process to twenty conjugate gradient iterations); we retained for testing that of three to five randomly initialized models which the evidence most favoured. In the case of the IVM, hyperparameter optimization must be interleaved with active set selection as described in Seeger et al. (2002, sec. 3.3).

Identical tests were run for a range of active set sizes on the IVM and SPGP classifier, and we have attempted to present the large body of results in its most comprehensible form: in this section, we list only the *sparsest* competitive solution obtained. This means that using a value for M smaller than shown tends to cause a marked deterioriation in performance, but it should not be inferred that there is no advantage in increasing the value. After all, as M approaches N we expect error rates to match those of the full model (at least for the IVM, which restricts itself to a subset of the training data). However, we believe that in exploring the behaviour of a sparse model, the essential question should be: what is the greatest sparsity we can achieve without compromising performance—since if sparsity were not an issue, we would probably use the full GP.

[8] Available from `http://www.stats.ox.ac.uk/pub/PRNN/`.
[9] Available from `http://ida.first.fhg.de/projects/bench/benchmarks.htm`.
[10] Carl Rasmussen's *gpml* package was used: `http://www.gaussianprocess.org/gpml/code/`.
[11] We used Anton Schwaighofer's code: `http://ida.first.fraunhofer.de/~anton/software.html`.
[12] Neil Lawrence's implementation was used: `http://www.cs.man.ac.uk/~neill/gpsoftware.html`.

Table 2.1: Test errors and predictive accuracy (smaller is better) for the GP classifier (GPC), the SVM, the IVM, and the sparse pseudo-input GP classifier (SPGPC).

Data set			**GPC**		**SVM**		**IVM**			**SPGPC**		
name	train:test	dim	err	nlp	err	#sv	err	nlp	M	err	nlp	M
synth	250:1000	2	0.097	*0.227*	0.098	98	0.096	*0.235*	150	**0.087**	*0.234*	4
crabs	80:120	5	0.039	*0.096*	0.168	67	0.066	*0.134*	60	**0.043**	*0.105*	10
banana	400:4900	2	0.105	*0.237*	0.106	151	**0.105**	*0.242*	200	0.107	*0.261*	20
breast-cancer	200:77	9	0.288	*0.558*	**0.277**	122	0.307	*0.691*	120	0.281	*0.557*	2
diabetes	468:300	8	0.231	*0.475*	**0.226**	271	0.230	*0.486*	400	0.230	*0.485*	2
flare-solar	666:400	9	0.346	*0.570*	**0.331**	556	0.340	*0.628*	550	0.338	*0.569*	3
german	700:300	20	0.230	*0.482*	0.247	461	0.290	*0.658*	450	**0.236**	*0.491*	4
heart	170:100	13	0.178	*0.423*	**0.166**	92	0.203	*0.455*	120	0.172	*0.414*	2
image	1300:1010	18	0.027	*0.078*	0.040	462	**0.028**	*0.082*	400	0.031	*0.087*	200
ringnorm	400:7000	20	0.016	*0.071*	0.016	157	0.016	*0.101*	100	**0.014**	*0.089*	2
splice	1000:2175	60	0.115	*0.281*	**0.102**	698	0.225	*0.403*	700	0.126	*0.306*	200
thyroid	140:75	5	0.043	*0.093*	0.056	61	0.041	*0.120*	40	**0.037**	*0.128*	6
titanic	150:2051	3	0.221	*0.514*	**0.223**	118	0.242	*0.578*	100	0.231	*0.520*	2
twonorm	400:7000	20	0.031	*0.085*	0.027	220	0.031	*0.085*	300	**0.026**	*0.086*	2
waveform	400:4600	21	0.100	*0.229*	0.107	148	0.100	*0.232*	250	**0.099**	*0.228*	10

Small values of M for the FITC approximation were found to give remarkably low error rates, and incremented singly would often give an improved approximating distribution. In contrast, the IVM predictions were no better than random guesses for even moderate M—it usually failed if the active set was smaller than a threshold around $N/3$, where the subset of data method was simply discarding too much information—and greater step sizes were required for noticeable improvements in performance. With a few exceptions then, for our model we explored a range of small M, while for the IVM we employed larger values which were more widely spread.

A more challenging classification problem is presented by the task of discriminating 4s from non-4s in the USPS database: the data are 256-dimensional, with 7291 training and 2007 test points. Provided with 200 pseudo-inputs (i.e. 51,200 parameters for optimization), error rates for our model are 1.94%, with an average NLP on the test set of 0.051 nats. These figures improve with 400 pseudo-inputs, to 1.79% and 0.048 nats. When provided with only 200 points, the IVM figures are 9.97% and 0.421 nats,[13] but given an active set of 400 its error rates are 1.54% and its NLP 0.085 nats.

[13]This can be regarded as a failure to generalize: it corresponds to labelling all test inputs as "not 4".

2.4 Discussion

The most arresting observation from many of these experiments is just how few pseudo-inputs the FITC approximation allows us to use, and how sparsely many of these old benchmark problems can be represented. Error rates are comparable and (perhaps surprisingly) occasionally superior to the full GP model, but require just a handful of kernel evaluations. In almost all cases, they improve on the IVM using only a fraction of the points required in the IVM active set: generally fewer than 10% of the inducing inputs required by the latter, and occasionally as few as 1% (see *diabetes*, *twonorm*). Furthermore, the predictive certainty is broadly preserved, giving us confidence that the FITC approximation does not only produce a sensible decision boundary, but is well able to model the underlying distribution.

Beyond yielding very fast classifiers, the minimum effective value M also reveals a great deal about the intrinsic complexity of our data, in a manner that is less apparent from, for example, the requisite size of active set in the IVM, or the number of support vectors used by the SVM. We return to this point below.

Superficially, the FITC approximation appears to be very similar to a semi-parametric family of models known as radial basis function (RBF) networks (Bishop, 1995). Indeed, by writing the mean prediction of FITC as a linear combination of kernel evaluations $\boldsymbol{\alpha}^T \mathbf{K}_{\bar{\mathbf{f}}\star}$ (see section 2.2.3), we recover exactly the form of prediction made by RBF nets. However, in its relationship to an underlying probabilistic model, FITC goes much further: first, we obtain estimates of variance in the latent signal at test points; second, there is the well-motivated evidence framework to guide optimization of kernel parameters and the placement of basis centres, the latter of which must be set to minimize some loss function in the RBF network, with the assumption that using sufficiently few basis functions will not allow overfitting.

This touches on an interesting issue: so far, we have fit all model parameters by maximizing the evidence, and we might ask if the same approach can also help us in choosing M itself. Although Quiñonero-Candela (2004) observes that the criterion is more reliable than an unregularized version due to Smola and Bartlett (2001), we find that the marginal likelihood has a tendency to fall monotonically with M. This is most easily understood by witnessing that we are only restoring the explicative power of the original GP, and always integrating the latent $\bar{\mathbf{f}}$; in other words, we do not introduce significant surplus flexibility that the framework penalizes. In this respect, FITC

is rather different from the RVM—which makes a curious hybrid of Bayesian principles and evidence optimization explicitly to eliminate basis functions—even though qualitatively, the active sets of each model appear to choose "representative" areas of the input domain, typically distant from decision boundaries.

We have already seen a similar structure to FITC in the projected process approximation, whose covariance consists solely of the low-rank term $\mathbf{PMP}^T$. In comparing their SPGP model with PLV, Snelson and Ghahramani (2006a) suggest that it is by the diagonal component in the FITC covariance, which corrects for the underestimated variance away from pseudo-inputs and restores the diagonal of the approximation to its true value, that the optimization of $\bar{\mathbf{X}}$ *by gradient ascent on the marginal likelihood* can succeed: without the noise reduction afforded locally by relocating pseudo-inputs, PLV does not provide a sufficiently large gradient for them to move, and the optimization gets stuck. We believe the same mechanism operates in general for non-Gaussian noise. This difficulty would not be significant if alternative heuristics for building the active set greedily were effective. We hypothesize however that in the classification domain, the most informative vectors in the greedy sense of the IVM tend to be those which lie close to the decision boundary.

We illustrate with a simple example that, provided the optimization is feasible, very sparse solutions may more easily be found if the inducing inputs can be positioned independently of the data. This allows the size of the active set to grow with the complexity of the problem, rather than the number of training points. We drew samples from a two-dimensional "xor" problem, consisting of four unit-variance Gaussian clusters at $(\pm 1.5, \pm 1.5)$ with a small overlap, giving an optimal error rate of around 13% and in loose terms a complexity which requires an active set of size four. By increasing the size of the training set N in increments from 40 to 400, we obtained the learning curves of fig. 2.3 for the IVM and FITC models: plotted against N is the size of active set required for the error rate to fall below 15%. Whereas the FITC model requires a constant four points to explain the data, the demands of the IVM appear to increase almost linearly with N.

Evidently, the FITC model is able to capture salient details more readily than the IVM, but we may object that it is also a richer likelihood. We therefore show learning curves for the FITC approximation run using the IVM active set and, generously, optimal kernel parameters. With a relatively simple and low-dimensional problem, the benefit of the adaptable active set that FITC offers is clearly much less significant than that of the

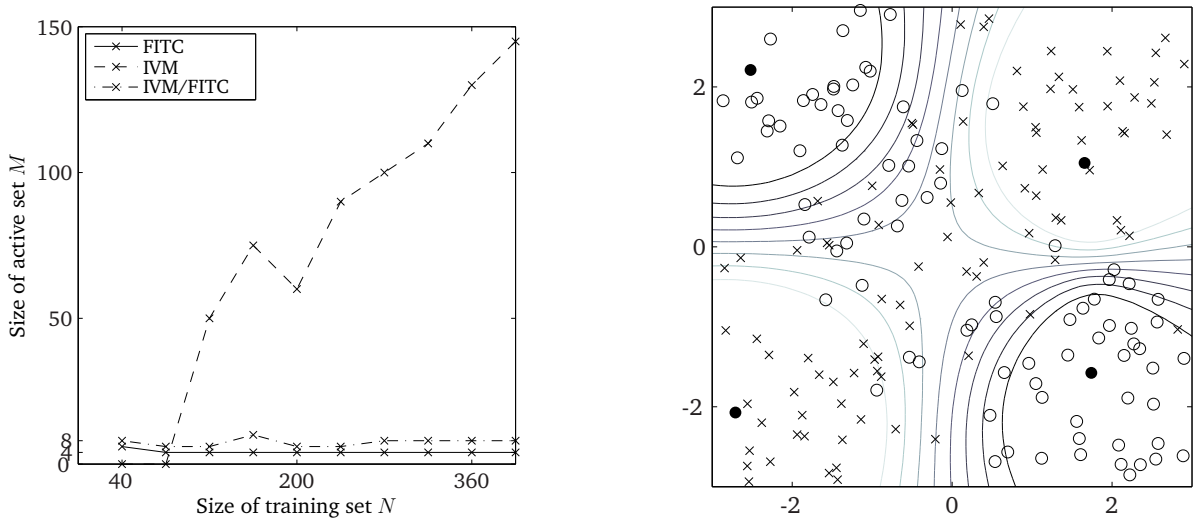


Figure 2.3: On the left-hand side, we plot learning curves for the toy problem described in the text. The second plot illustrates contours of posterior probability obtained for FITC in ten conjugate gradient iterations from a random initialization of pseudo-inputs. These appear as black dots towards the centre of the generative clusters.

improved approximation itself; with a handful more basis functions (as chosen by the IVM), FITC succeeds in learning the decision boundary. However, in two dimensions, the task is perhaps too simple; therefore, we also consider how the growth of M is related to the dimensionality D with a similar set of tests for more complex exclusive-or style problems, generating data from 2^D isometric Gaussian distributions with small overlap. In the first case, we used 80 training inputs in two dimensions, similar to before; in the second case we increased the dimensionality to 3, and the number of training inputs proportionately to 160; in the final case, we used 320 points in four dimensions, always distributing the data evenly between the quadrants: the complexity of the problem grows exponentially in D. The results appear in table 2.2, averaged over five data sets. In the FITC case we present the best model of three, in evidential terms, obtained after thirty CG iterations. It is clear that as D grows, the greediness of the IVM becomes rapidly sub-optimal, even when the richer noise model of FITC is used after the active set has been chosen. In contrast, the continuous optimization of SPGP classification lets M scale more in accordance with the complexity. It is interesting to note how the full GP appears to overfit; by constraining the covariance through the FITC approximation, we achieve a slight improvement in generalization. This effect may not be the fluke of an artificial setup, occurring in some benchmark tests above (*synth*, *thyroid*) where FITC was also superior to the full GP.

Table 2.2: Contrasting FITC and IVM.

Dimension	**GP**	**IVM**		**IVM/FITC**		**SPGPC**	
		M	err	M	err	M	err
2	0.069	4	0.406	4	0.321	**4**	**0.054**
		6	0.322	6	0.173	6	0.055
		8	0.287	8	0.134	8	0.056
		16	0.135	16	0.067		
		32	0.061	24	0.062		
3	0.090	8	0.565	8	0.418	8	0.122
		12	0.534	12	0.276	**12**	**0.083**
		16	0.449	16	0.173	16	0.082
		32	0.286	32	0.092		
		96	0.088	64	0.087		
4	0.139	16	0.484	16	0.406	16	0.233
		24	0.482	24	0.345	**24**	**0.133**
		32	0.543	32	0.244	32	0.121
		192	0.196	64	0.150		
		256	0.138	96	0.130		

Although we believe it is ultimately a more accurate approach, the principal problem with using gradient ascent to locate the pseudo-inputs is the increased training times. A sensible compromise can be reached when the full optimization is unfeasible by greedily obtaining the active set, but switching to the FITC approximation for optimization of kernel parameters, or only optimizing a small selection of the pseudo-inputs. In the next section we consider an alternative idea for reducing the computational burden.

2.5 Dimensionality reduction

For problems on a very large scale, such as the USPS task attempted above, the non-linear optimization is extremely demanding, involving tens of thousands of parameters. If the data are D-dimensional, there are MD parameters to learn for the M inducing inputs, in addition to hyperparameters of the kernel. We suggested earlier how we might employ more intelligent initialization strategies; a second option is to optimize only a manageable subset of the pseudo-inputs.

Another approach altogether is explicitly to reduce the dimensionality of the data, by projecting them into a d-dimensional subspace. In this case, all kernel evaluations involving the inducing inputs require their projections also, hence we can treat the $\bar{\mathbf{X}}$ as existing only on the low dimensional manifold. There are an extra Dd parameters to learn for the projection matrix, but only Md for the pseudo-inputs. Also, there is probably a requirement for fewer kernel hyperparameters, since automatic relevance determination effects theoretically can be achieved directly through adjustments to elements of of the projection matrix. This approach was applied to the tractable FITC model in Snelson and Ghahramani (2006b), having already appeared in the context of the full GP in Vivarelli and Williams (1999).

To make an optimal projection (in the type-II ML sense), there remains the question of efficient calculation of the gradient of the marginal likelihood with respect to these elements P_{ij}, where i ranges over the dimensions $1 \dots D$, and j over dimensions $1 \dots d$. Observe that $\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}$ is independent of the projection, since by construction the pseudo-inputs already exist on the manifold. If we consider only stationary covariance functions, for which $K(\mathbf{x}, \mathbf{x})$ is constant, then $\nabla_{P_{ij}} \mathsf{diag}\left(\mathbf{K}_{\mathbf{ff}}\right)$ is also constant, and to apply the SPGP model we are left with the task of calculating

$$\frac{\partial \mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}}{\partial P_{ij}} = \left.\frac{\partial \mathbf{K}(\bar{\mathbf{X}}, \mathbf{Z})}{\partial \mathbf{Z}}\right|_{\mathbf{Z}=\mathbf{XP}} \frac{\partial \mathbf{XP}}{\partial P_{ij}}, \tag{2.11}$$

where we have arranged data $\mathbf{X}$ in a matrix of size $N \times D$, so that their projection is the matrix $\mathbf{XP}$. The first partial derivative is a tensor, most easily visualised as a matrix of $N \times d$ entries (corresponding to the Nd elements of $\mathbf{XP}$), each of which is itself an $M \times N$ matrix of partial derivatives. The second term is a matrix $d \times N$, but of which only the jth row is non-zero, equal to $\mathbf{X}^T_{(\cdot,i)}$, i.e. the ith coordinate of the unprojected data. Hence, since it appears in product with this matrix, we are interested only in the jth column of the tensor (see fig. 2.4).

Consider this jth column in greater detail: it consists of N matrices, the partial derivatives of $\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}$ w.r.t. a change in the jth coordinate of each projected datum. The consistency of GPs ensures that for the nth such matrix, only the nth column can be non-zero (moving the nth datum cannot affect covariance evaluations involving other data). For maximum efficiency then, this entire tensor column of matrices can safely be compressed into a single matrix ∇_j. The solution to (2.11) becomes the Hadamard or element-wise product of ∇_j with M repeated rows of $\mathbf{X}^T_{(\cdot,i)}$.

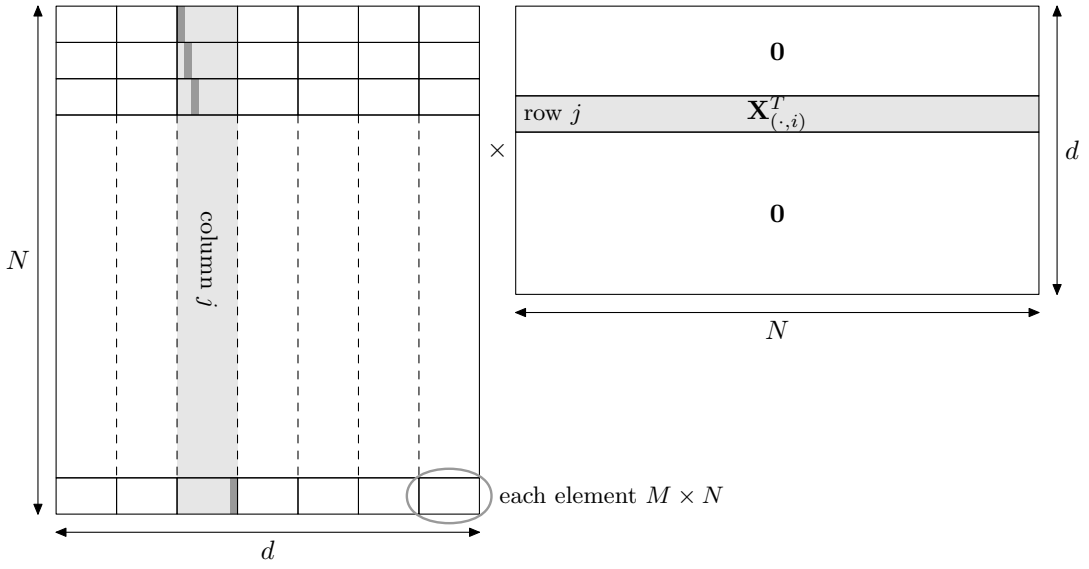


Figure 2.4: Representation of the tensor product (2.11) involved in calculating derivatives of the log marginal likelihood with respect to elements of the projection matrix P_{ij}. The right-hand matrix selects only column j on the left-hand side, of which only the individually highlighted columns are non-zero.

2.5.1 The isotropic squared exponential

For the isotropic squared exponential kernel (1.3) (omitting the lengthscale parameter l), and by writing $\tilde{\mathbf{x}}^T = \mathbf{x}^T\mathbf{P}$, the necessary derivative is readily found to be

$$\frac{\partial K(\bar{\mathbf{x}}, \tilde{\mathbf{x}})}{\partial \tilde{x}_j} = K(\bar{\mathbf{x}}, \tilde{\mathbf{x}})\,(\bar{x}_j - \tilde{x}_j)\,,$$

such that

$$\nabla_j = \mathbf{K}_{\bar{\mathbf{f}}\tilde{\mathbf{f}}} \bullet \mathsf{lindist}\left(\bar{\mathbf{X}}_{(\cdot,j)}, \tilde{\mathbf{X}}_{(\cdot,j)}\right),$$

where $\bullet$ denotes a further Hadamard product, and $\mathsf{lindist}(\mathbf{u}, \mathbf{v})$ for vectors $\mathbf{u}$ and $\mathbf{v}$ is a matrix of all pairwise distances $u_m - v_n$. We reiterate that an anisotropic kernel is not required since variations of lengthscale are possible through adjustment of $\mathbf{P}$.

Calculating the gradients $\frac{\partial L}{\partial P_{ij}}$ efficiently remains slightly arduous if we are to avoid any N^2 complexity. However, since the projection does not affect the inducing inputs, many terms in the derivative equations evaluate to zero; further details are provided in appendix B.

2.5.2 Experiments

We conducted experiments on the image, splice and USPS sets, some of the higher-dimensional data we used in the main experiments section. Unfortunately, we must report the failure of our algorithm, since in all cases but image (results in table 2.3) were the results extremely poor. Indeed, even for the image set, error rates are far from competitive with those obtained by the methods in table 2.2.

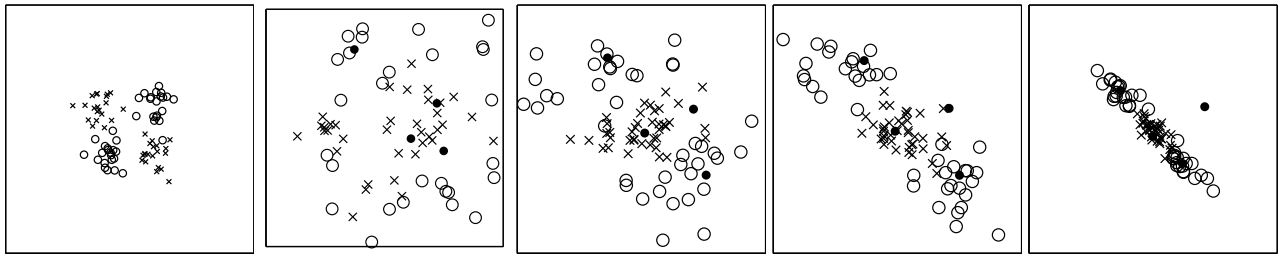
Table 2.3: Results for learning low-dimensional projections. In the original experiments, there were $200 * 18 + 2 = 3602$ hyperparameters for the image set.

Data set	**Dimension**	M	**No. hyperparameters**	err	nlp
image	2	30	67	0.0782	0.242
	4	30	103	0.0688	0.160
	6	30	139	0.0995	0.203
	8	30	175	0.1520	0.304

2.5.3 Discussion

The results appear to indicate that, at least for binary classification, learning a projection of the data may not simplify the problem. There are several difficulties with the method, and we mention first an issue arising at the level of implementation. It turns out that derivative calculations are rather slow for elements of $\mathbf{P}$: in the original model, when the position of a pseudo-input is optimized, the gradients for all D components may be calculated in parallel. No such vectorization applies for the projection matrix because there is no independence property between its elements. As a result, tuning the matrix is disproportionately slow with respect to the number of model parameters since they are iterated over in turn.

Second, even given the necessary time for training, disruptive local optima in the evidence are often revealed as we tune the projection. To illustrate the problem, we refer to the following experiment. Data were drawn from a two-dimensional exclusive-or problem to which an extra eight dimensions were introduced consisting of pure noise, i.i.d. $\mathcal{N}(0, 1)$. We compare the performance of a method which projects back into 2-d (requiring 30 hyperparameters) with that of a full FITC model and anisotropic kernel (52 hyperparameters), in both cases providing only the four pseudo-inputs theoretically required to solve the problem.



(a) Original data (b) 6 iterations (c) 14 iterations (d) 16 iterations (e) 30 iterations

Figure 2.5: These data are nearly separable in two dimensions, but the optimizer has found a sub-optimal, essentially unidimensional projection. We chart its progress in terms of the number of CG iterations. The solid black dots mark the pseudo-inputs; observe that in the final panel, one has become divorced from the data (although, for the chosen projection, only three are required).

The optimal solution clearly is to disregard the extra dimensions, in the case of the projection by giving them zero weight, and for the ARD kernel, by associating with them very large lengthscales. In the latter case, this happens quite reliably; typically, after thirty conjugate gradient iterations the mean log lengthscale on the corrupting components is around 3, that on the two useful components around zero, and the pseudo-inputs are usually learned effectively, yielding a near-optimal classifier. In thirty iterations the projection model also converges, and very often to a solution in which the four largest elements of the projection matrix are those affecting the signal.

There are three problems however; first, the remaining elements of the projection are driven to zero only slowly (their r.m.s. value after the thirty iterations is around 0.2); second, in the joint optimization, pseudo-inputs can become divorced from real data, essentially pruning a basis function; third, and perhaps most fundamental, is the inferiority of the projection itself, which usually concentrates on only the first component of variance, that which extends diagonally through the origin, causing an unnecessary class overlap: the optimizer pursues a local valley along which the data are increasingly compressed on one axis. This helps to explain the separation of real data from pseudo-inputs during the optimization, since once the data have been projected down to just one dimension, only three pseudo-inputs are strictly necessary to achieve the optimal error, and the gradients on the fourth are evidently insufficient to move it further. Fig. 2.5 illustrates the progress of the algorithm by plotting training data and pseudo-inputs in the low-dimensional space during the course of a typical optimization.

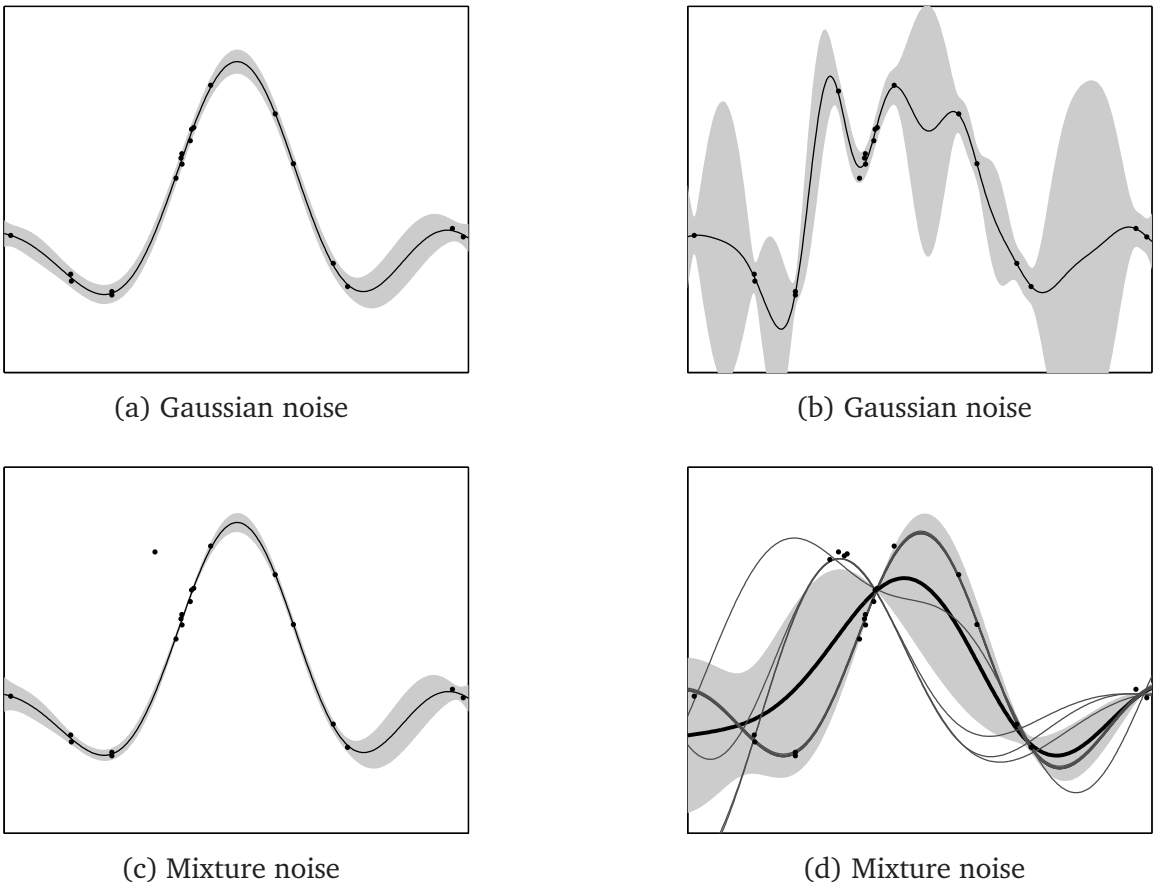
It is true that with a more fortunate initialization, the optimizer can recover a better projection, but equally, with a less fortunate initialization, it makes almost no progress at all: over ten data sets, its minimum error rate was 7.5%, with an average of 37%; for the ARD kernel, these figures are 2% and 6.8%. Furthermore, training for the latter was slightly quicker, by virtue of the inefficient gradient calculations in the projection model. Each of the difficulties we identify here can only grow more severe with the number of dimensions, and we are forced to conclude that the very largest problems may simply not be amenable to a global optimization strategy like SPGP, but must resort at least in part to the greedy growth of an active set.

CHAPTER 3

Robust Gaussian process regression

NATURALLY OCCURRING REGRESSION data are often modelled as noisy observations of an underlying function. The conventional assumption is that all noise is i.i.d. zero-mean Gaussian, such that a typical set of samples appears as a cloud around the latent function. Gaussian processes are well-suited to these conditions, for which all computations remain tractable (see fig. 3.1a).

In the context of GPs, the Gaussian noise model enjoys computational advantages. There is also the theoretical justification of the *central limit theorem*, which states that the sum of sufficiently many i.i.d. random variables of finite variance will be distributed normally. However, only rarely can perturbations affecting data in the real world be argued to have originated in the addition of many i.i.d. sources. The random component in the signal may be caused by human or measurement error, or it may be the manifestation of systematic variation invisible to a simplified model. This means that an "outlier" may or may not be a genuinely erroneous measurement, and can only be viewed as such with reference to the modelling assumptions embodied in the sampling distribution. If outlying observations are common they could be indicative of model mismatch. In any case, if ever there is the possibility of encountering relative to our model small quantities of highly implausible data, we require *robustness*, i.e. one whose predictions are not grossly affected by large errors.



(a) Gaussian noise

(b) Gaussian noise

(c) Mixture noise

(d) Mixture noise

Figure 3.1: Black dots show noisy samples from the `sinc` function. In panels (a) and (b), the behaviour of a GP with a Gaussian noise assumption is illustrated; the shaded region shows 95% confidence intervals. The presence of an outlier is highly influential here, but the heavy-tailed likelihood (3.1) in panel (c) is more resilient. Even this model fails for the cluster of outliers in panel (d), where grey lines show the means of ten repeated runs of the EP inference algorithm, and the black line and shaded region indicate their mean and variance—grossly at odds with those of the latent generative model.

For simple univariate data, it is sometimes possible to screen manually for outliers, but in the multivariate case the latent distribution is usually sufficiently obscure that this will be impossible; regardless, it is wasteful to use humans as preprocessors for an inference procedure that is otherwise automated. It can also be wasteful or even harmful to discard outliers entirely when we may need only to moderate their influence or flag their presence: their removal inevitably affects the statistics of the data set, for example leading to underestimation of the variance.

Demands for robustness render the standard GP inappropriate: the light tails of the Gaussian distribution cannot explain large non-Gaussian deviations, which either skew

the mean interpolant away from the majority of the data, or force us to infer an unreasonably large (global) noise variance; these effects are illustrated in fig. 3.1b. Robust methods use an i.i.d. heavy-tailed likelihood to allow the interpolant effectively to favour smoothness and ignore such erroneous data: fig. 3.1c shows how this can be achieved using the mixture of Gaussians noise model described in section 3.1.2.

In this chapter, we address the more fundamental GP assumption of i.i.d. noise. Our research is motivated by observing how predictions suffer for heavy-tailed models when outliers appear in bursts: fig. 3.1d replicates fig. 3.1c, but introduces an additional three outliers. All hyperparameters were taken from the optimal solution to (c), but even without the challenge of their optimization, there is now considerable uncertainty in the posterior since the competing interpretations of the cluster as signal or noise have similar posterior mass. Viewed another way, the tails of the *effective* log likelihood of four clustered observations have approximately one-quarter the weight of a single outlier, so the magnitude of the posterior peak associated with the regularized solution is comparably reduced.

It is not clear how to treat the several peaks in the posterior, some of which are "correct", others of which correspond to spurious interpretations of the data. We illustrate the various optima discovered by EP, naïvely plotting the mean of ten repeated runs (with a randomized order for site refinement) to emphasise the multimodality of the posterior as well as the instability of the algorithm. Of course, this averaging cannot be justified as a sound inference procedure: in general, the different modes of the posterior are comparable only via a Bayesian posterior weight.

One simple remedy to the problem of poor convergence is to make the tails of the likelihood heavier. However, although we may be able to establish a globally optimal likelihood distribution by gradient ascent on the evidence, this choice will have ramifications across the entire data space. It is possible that no single noise model will be satisfactory everywhere since the tails may be too heavy in some regions (causing underfitting when real data are explained as outliers), and too light in others (when outliers cause an undesirable kink in the predictive mean). The ideal solution would be a noise model whose predictive variance could itself vary in the input domain: in this chapter, we introduce just such a model. By applying a GP gating function to partition the domain softly into "real" and "outlier", the noise distribution varies in an input-dependent manner such that in regions of confidence, the tails can be made very

light (encouraging the interpolant to hug the data points tightly), while more dubious observations can be treated appropriately by broadening the distribution in their vicinity.

In the next section we describe briefly the frequentist perspective on robustness, and detail some common methods for robust GP regression, followed in section 3.2 by a description of our new model, the twinned GP. Section 3.3 presents some experimental results, while a Monte Carlo algorithm for sampling from the posterior appears in section 3.4. We conclude in section 3.5 with an evaluation and discussion of the twinned GP in the context of related approaches in the literature.

3.1 Classical methods

In this thesis, we are interested in a Bayesian concept of robustness of inference. However, for completeness, we begin with a short review of the robust estimation of criteria from data.

3.1.1 Robust estimators

The frequentist literature considers robustness with respect to the estimation of various statistics of a sample. As a motivating example, consider the sample mean $\sum_{n=1}^{N} y_n$ which has a "breakdown point" of 0%—that is, it can be made arbitrarily large by pushing a single observation towards $\pm\infty$. In contrast, the median is resilient to such corruptions; indeed, with a breakdown point of 50%, fully half of the data may be corrupted before the median can be made arbitrarily large.

The mean and median are examples of *M-estimators* (Huber, 1981)—of which only the latter is robust—one of a variety of classes of estimator that arise as the solution to a particular optimization problem. It generalizes the maximum likelihood solution

$$\hat{\theta} = \arg\max_{\theta} \prod_{n=1}^{N} f(y_n|\theta) = \arg\min_{\theta} \left(-\sum_{n=1}^{N} \log\left(f(y_n|\theta)\right) \right)$$

to a wider class of functions ρ:

$$\hat{\theta} = \arg\min_{\theta} \left(\sum_{n=1}^{N} \rho(y_n|\theta) \right),$$

where the function ρ is chosen for certain desirable properties, typically to moderate or eliminate the influence of outlying observations. The optimization is usually solved by an iterative algorithm (*IRLS* or iteratively reweighted least squares), but for many interesting choices of ρ there may be multiple solutions, and initialization is important.

Concrete examples are given by the mean, for which $\rho(y_n|\mu) = (y_n - \mu)^2$, and the median, which corresponds to $\rho(y_n|m) = |y_n - m|$. Other robust estimates of the mean involve truncating the "loss" outside a given range $\pm c$ (called "trimming"), or restricting its growth to a linear function of $y_n - \mu$ ("Winsorizing"); for more examples, see Huber (1981) and Rousseeuw and Leroy (1986).

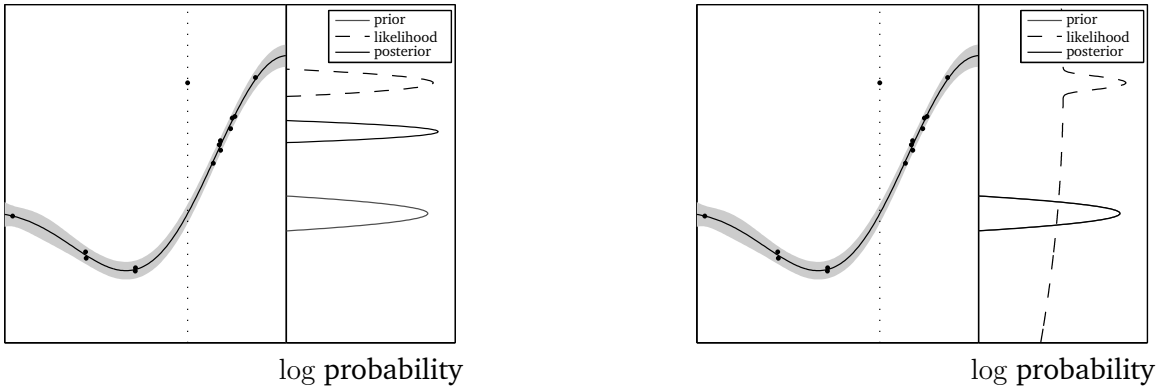
A general algorithm for estimation of the parameters of a mathematical model is *RANSAC* (for random sample consensus), given by Fischler and Bolles (1981). It assumes most of the data are "inliers", from which the parameters may be determined reliably. By repeated resampling of subsets of the data, the method determines which partitions are most reliable by observing how well the remaining data are explained by a model fit on the sampled set. In the spirit of frequentism, only the parameters of the best supported model are retained; those from the more dubious are simply discarded.

3.1.2 Robust GP regression

Robust GP regression is achieved by using a *leptokurtic* likelihood distribution, i.e. one whose tails have more mass than a Gaussian (Box and Tiao, 1973, sec. 3.1.1). The mechanism is most easily understood with reference to a new observation (in EP terms, we would speak of a *site inclusion*) which fails to conform to our expectations;[1] see fig. 3.2. The updated posterior is proportional to the product of this "prior" and the likelihood term: it is pulled strongly towards the observation if the tails of the likelihood are light; only by making them relatively heavier can the influence of the prior survive into the posterior.

The proportion of mass in the tails relative to the peak determines how readily the posterior can drift from the data without incurring a strong evidence penalty: if the tails are very light, signals that are distant from corrupted observations have vanishingly small posterior probabilities, hence data with outliers demand a short lengthscale

[1] That is, of our beliefs immediately before the observation, proportional to the product of a Gaussian prior and a series of heavy-tailed likelihood distributions for the data observed so far—a product usually approximated by a Gaussian distribution, and which in Bayesian terms represents a prior or contextual belief.



(a) Gaussian noise (b) Mixture noise

Figure 3.2: The left-hand plot in each figure shows an enlargement of fig. 3.1a. The effect on the marginal posterior at the dotted line of introducing the outlier in fig. 3.1b is illustrated, in (a) for a Gaussian likelihood, and in (b) for a mixture of two Gaussians. In the latter case, the posterior is essentially indistinguishable from the prior, as desired.

or large global noise variance to allow the signal to explain those errant observations satisfactorily. Heavy tails afford significant probability to signals which ignore outliers and respect local structure, and may favour a posterior belief in smoother, low-noise predictions.

Three popular heavy-tailed distributions are compared in fig. 3.3:

$$\text{the mixture of Gaussians} \quad p(y|f;e,\sigma_R^2,\sigma_O^2) = (1-e)\mathcal{N}\left(y\,;\,f\,,\,\sigma_R^2\right) + e\mathcal{N}\left(y\,;\,f\,,\,\sigma_O^2\right); \tag{3.1}$$

$$\text{Student's t distribution} \qquad p(y|f;\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{(y-f)^2}{\nu}\right)^{\left(-\frac{\nu+1}{2}\right)}; \tag{3.2}$$

$$\text{the Laplace distribution} \qquad p(y|f;\lambda) = \frac{1}{2\lambda}\exp\left(-\frac{|y-f|}{\lambda}\right). \tag{3.3}$$

Their parameters (e, the prevalence of outliers, and σ_R^2, σ_O^2, the variance of each mixture component; ν, the number of degrees of freedom; and λ, the so-called "rate" parameter) can all be viewed as varying the kurtosis. The mixture of Gaussians has been suggested by Box and Tiao (1968); it was also advocated by Jaynes (2003, ch. 21) as a "two-model model": it explicitly separates the kinds of corruption expected for trusted and outlying observations. One benefit of this distinction is the straightforward infer-

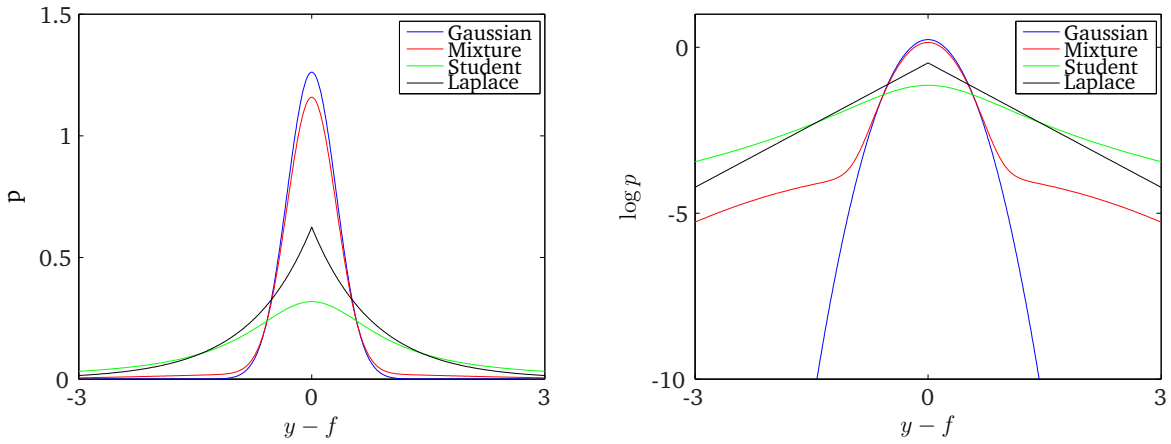


Figure 3.3: The p.d.f. for three heavy-tailed distributions and the (light-tailed) Gaussian.

ence of only the normal observation process. All three distributions can be written in the form of a scaled mixture of Gaussians,

$$p(y|f,\boldsymbol{\theta}) = \int \mathcal{N}\left(y\,;\,f\,,\,\sigma^2\right) q(\sigma^2|\boldsymbol{\theta}) d\sigma^2, \tag{3.4}$$

as introduced by Andrews and Mallows (1974). In particular, if q is inverse Gamma, we recover Student's t distribution (3.2); if q is exponential, we recover the Laplace distribution (3.3); the finite mixture (3.1) is clearly a degenerate case.

Inference requires the calculation of

$$p(\mathbf{f}|\mathbf{X},\mathbf{y}) = \frac{p(\mathbf{f}|\mathbf{X})p(\mathbf{y}|\mathbf{f})}{\int p(\mathbf{f}|\mathbf{X})p(\mathbf{y}|\mathbf{f})\mathrm{d}\mathbf{f}} = \frac{p(\mathbf{f}|\mathbf{X})\prod_{n=1}^{N} p(y_n|f_n)}{\int p(\mathbf{f}|\mathbf{X})\prod_{n=1}^{N} p(y_n|f_n)\mathrm{d}\mathbf{f}}.$$

The i.i.d. assumption means that the posterior is proportional to the product of a Gaussian prior and N heavy-tailed likelihood terms, but to normalize we need to marginalize over the latent $\mathbf{f}$. Unfortunately, in the case of (3.1) the calculation contains an exponential number of terms, and for (3.2) the integral does not even have a closed form. Use of the double exponential (3.3) yields in the log domain the sum of a quadratic form and N axis-aligned $\mathcal{C}_0$-continuous linear terms. The posterior as a function of $\mathbf{f}$ then consists of $\mathcal{O}(2^N)$ disjoint regions, each of which is a rescaled Gaussian distribution formed from the product of the Gaussian and one tail from each Laplace. Again, this exponential complexity demands an approximation. Kuss (2006) describes how to perform approximate inference in all three cases, for the mixture of Gaussians by EP and Markov chain Monte Carlo (MCMC) methods, for the Laplace distribution by

EP, and for Student's t distribution both by a variational approach and by an MCMC method that exploits the representation (3.4) and involves the explicit sampling of individual variances. EP presents a much faster alternative to stochastic methods, and for the mixture model and Laplace likelihoods, where the marginal moments are analytic, the update equations have a simple closed form, making the algorithm especially attractive.

Of these three distributions only the Laplace is log concave, a constraint that effectively limits the amount of mass a distribution can push from its "shoulders" into its tails. Indeed, the Laplace is the heaviest-tailed of all log concave distributions, and is the only heavy-tailed likelihood in common use to guarantee a unimodal posterior. This is of interest because multimodality raises certain practical difficulties with EP: Kuss (2006) finds more reliable convergence with the Laplace than the mixture. However, as discussed by Narula and Wellington (1982), its tails are still sufficiently light that a single observation can have an arbitrarily large effect on the posterior.

3.2 Twinned Gaussian processes

Consider the mixture noise (3.1). Each of the 2^N interpretations of the data is a partition into "real" and "outlier" classes, corresponding to a possible local optimum in the posterior distribution. How we resolve this multimodality depends on our method of approximate inference. Monte Carlo algorithms attempt directly to average over the full posterior, and uncertainty in the interpretation is reflected in the different samples. Deterministic methods (EP, VB) in contrast use a unimodal (Gaussian) approximation whose width can simulate this uncertainty only indirectly. They can be sensitive to initialization, and EP in particular is sensitive to the order in which the observations are considered. In this case, the rival interpretations are problematic since they disrupt convergence of the algorithm. Furthermore, different solutions generally have associated with them inconsistent derivatives of the evidence with respect to hyperparameters, presenting serious challenges for model selection. Ideally, we would like to mitigate the effects of these competing perspectives by somehow adjusting the relative magnitudes of the peaks in the posterior distribution: in a sufficiently rich model, knowledge of clustering behaviour or other correlations in the noise component could be incorporated into the inference process. Unfortunately, with the i.i.d. assumption of prevalent robust methods, it is very difficult to employ such knowledge since there is no possibility of local variation in noise.

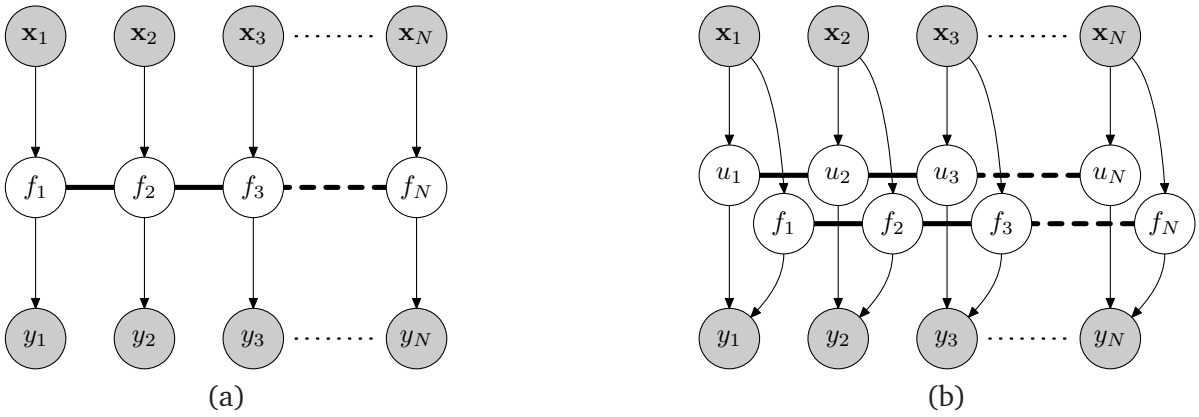


Figure 3.4: In panel (a) we show a graphical model for the Gaussian process. The data ordinates are $\mathbf{x}$, observations $\mathbf{y}$, and the GP is over the latent $\mathbf{f}$. The bold black lines indicate a fully-connected set. Panel (b) shows a graphical model for the twinned Gaussian process, in which an auxiliary set of hidden variables $\mathbf{u}$ describes the behaviour of noise on the data.

We now describe a model motivated by the shortcomings of existing deterministic solutions for robust GP regression. We have called it the *twinned Gaussian process* (TGP) by virtue of its graphical representation (fig. 3.4b). In fact, viewed in this way, the connectivity of the model is not novel,[2] but our noise model and implementation are new, allowing a posterior that avoids cumbersome Monte Carlo methods.

We augment the standard process over $\mathbf{f}$ with another over a set of variables $\mathbf{u}$, whose values probabilistically partition the domain into real and outlier components. The methodology is closely related to GP classification, in which a latent process is passed through a sigmoidal function to obtain a Bernoulli-distributed class label. Similarly here, the latent u_n is passed through the cumulative Gaussian $\sigma(\cdot)$ to give the probability that an observation is "real". In the generative model, we toss a suitably-weighted coin and draw the observation y_n from the relevant component. The priors on $\mathbf{f}$ and $\mathbf{u}$ are not constrained to be equal, reflecting the possibility that we hold quite different beliefs about correlations within the signal and the noise. In addition, we permit a non-zero mean process on $\mathbf{u}$:

$$p(\mathbf{u}|X) = \mathcal{N}(\mathbf{u}\,;\,\mathbf{m_u}\,,\,\mathbf{K_{uu}})\,, \qquad \text{and} \qquad p(\mathbf{f}|X) = \mathcal{N}(\mathbf{f}\,;\,\mathbf{0}\,,\,\mathbf{K_{ff}})\,.$$

[2]Goldberg et al. (1998) use a similar design, explicitly modelling the variances across the data, leading to an intractable posterior that must be sampled by Monte Carlo methods.

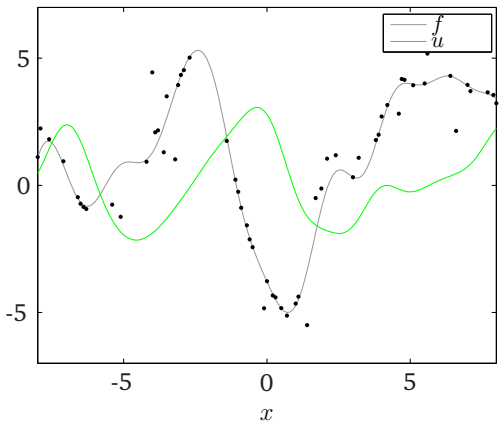


Figure 3.5: Data drawn from the marginal likelihood of the TGP.

Except where explicitly stated otherwise, we will assume $\mathbf{m_u} = [\, m_u \; m_u \; \cdots \; m_u \,]^T$, a constant vector encoding a uniform prior belief in the corruptibility of data.

3.2.1 The likelihood

The precise form of the likelihood $p(y_n|u_n, f_n)$ may be tailored to meet the requirements of the application domain. We give by way of example the natural generalization of the standard mixture model (3.1); more elaborate alternatives are explored in chapter 4. In this likelihood, two forms of Gaussian corruption are mixed, one strongly peaked at the observation, the other broader to provide the heavy tails. As mentioned above, instead of mixing these components in fixed proportions, we pass u_n through a sigmoid to obtain the probability of each component having generated the data. This makes intuitive sense, but equally important, it retains the advantage of tractability with respect to EP updates.

By assumption, y_n is a Gaussian corruption of f_n: with probability $1 - \sigma(u_n)$ it is an outlier, distorted by a large variance σ_O^2, while with probability $\sigma(u_n)$ the uncertainty is due to the small jitter on real data σ_R^2:

$$y_n = \begin{cases} \mathcal{N}(y_n\,;\, f_n\,,\, \sigma_O^2) & \text{with probability } e_n = 1 - \sigma(u_n), \\ \mathcal{N}(y_n\,;\, f_n\,,\, \sigma_R^2) & \text{with probability } 1 - e_n = \sigma(u_n). \end{cases} \tag{3.5}$$

For illustrative purposes, fantasy data generated from the model appear in fig. 3.5. Observe that in terms of generative ability the model is simplistic: there is no fine-grained control of variance, only a binary "real"/"outlier" switch. However, in the

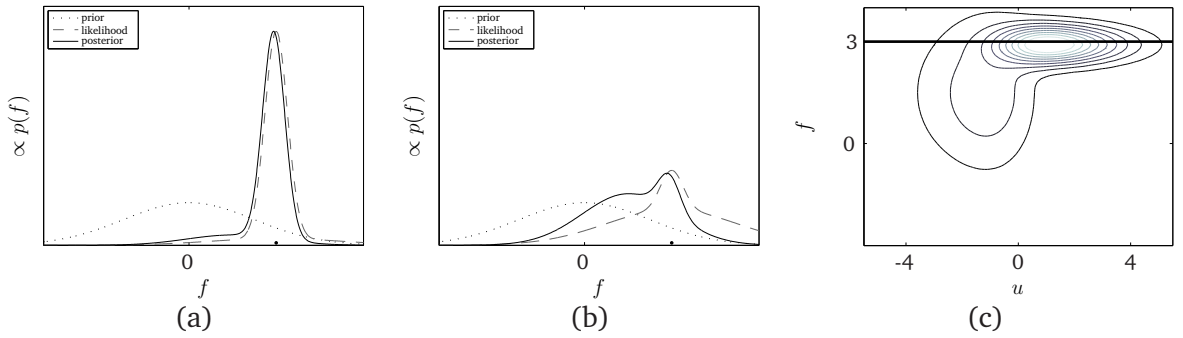


Figure 3.6: In panels (a) and (b), the dotted line is the prior distribution over latent f; the likelihood (dashed) is shown as a function of f given the observation denoted by a black dot; the resultant posterior over f is solid. The likelihood mixture is weighted to favour the real component in (a); in (b), the outlier component receives greater weight. Each panel shows a slice through the contours at $u = \pm 1$ of (c), which illustrates the posterior over u and f after observing the single datum denoted by the thick black line on the f-axis.

inference process, we are given a data set and a posteriori cannot be sure about the latent assignment. Our hope is that this uncertainty will appear in marginal variances ranging continuously from σ_R^2 to σ_O^2 (and up to the variance of the prior).

The TGP likelihood may be understood as lying between two extremes: the mixture (3.1) is recovered by forcing absolute correlation in $\mathbf{u}$ and adjusting the mean of the $\mathbf{u}$-process to $m_u = \sigma^{-1}(1 - e)$. Conversely, if all correlations in $\mathbf{u}$ are removed then at each input the assignment of responsibility between the two components is performed independently, i.e. u_n is drawn from the prior: the model reduces to a classic mixture, more commonly tackled with expectation maximization or variational methods. Between these poles the TGP uses the flexibility of a GP on $\mathbf{u}$ in effect to adapt e in an input-dependent manner (see fig. 3.6): the likelihood ranges from a sharp peak appropriate for regions of confidence, to a broader distribution more suitable for regions of suspect data.

3.2.2 Inference

The TGP requires we maintain the joint posterior over both $\mathbf{f}$ and $\mathbf{u}$, and although their priors are independent, we expect correlations to arise after conditioning on observations. To understand this, consider a single datum $(\mathbf{x}_n, y_n)$. In principle, it admits two explanations corresponding to its classification as either "real" or "outlier": in general terms, either $u_n > 0$ and $f_n \approx y_n$, or $u_n < 0$ and f_n respects the local structure of the signal (fig. 3.6c).

Since the likelihood involves a mixture of Gaussians, the true posterior distribution is exponentially complex and for all but the smallest problems approximate methods are required. In the literature, approximate inference for GP classification has involved stochastic sampling (Neal, 1997), variational methods (Gibbs and MacKay, 2000), and expectation propagation (Kuss and Rasmussen, 2005), as well as earlier approaches related to EP and drawn from a background of statistical physics (Opper and Winther, 2000). Rather outdated is Laplace's approximation, which fits a Gaussian to match the curvature at a local maximum in the posterior. This extreme value tends to be unrepresentative of the bulk of the posterior mass by virtue of the soft partitioning made by the sigmoidal likelihoods. Kuss and Rasmussen demonstrate the inferiority of Laplace's method to EP, corroborating theoretical arguments in favour of the latter's global perspective; indeed, extensive comparative tests presented in their paper allow a succinct summary: for binary GP classification, EP is the method of choice in terms of both speed and accuracy. Of paramount importance to our algorithm is speed of inference, and with this endorsement we have chosen an EP-based inference procedure. We also assess the faithfulness of the approximation to the true posterior in section 3.4, by drawing samples from the latter using MCMC.

3.2.3 Implementation

We outlined a generic algorithm for EP in section 1.3. In the TGP we desire a posterior approximation over both $\mathbf{f}$ and $\mathbf{u}$, and to this end must maintain the entire distribution $\mathcal{N}([\mathbf{u}^T\,\mathbf{f}^T]^T; \mathbf{h}, \mathbf{A})$ during the course of the EP iterations. The space requirements of the covariance matrix are $\mathcal{O}(4N^2)$, and that of the mean vector $\mathcal{O}(2N)$. The site approximations, unnormalized and rescaled Gaussian distributions with natural parameters $\mathbf{b}_n$ and $\mathbf{\Pi}_n$, are local to the nth observation, and thus each $\mathbf{b}_n$ is a vector length 2, and each $\mathbf{\Pi}_n$ a 2×2 symmetric matrix. These can be encoded together with the N scale parameters z_n in $\mathcal{O}(6N)$ space.

The algorithm is initialized at the prior, i.e. the approximate posterior

$$\mathcal{N}\left(\begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}; \mathbf{h}, \mathbf{A}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}; \begin{bmatrix}\mathbf{m_u}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{K_{uu}} & \mathbf{0}\\ \mathbf{0} & \mathbf{K_{ff}}\end{bmatrix}\right),$$

and for the sites, all parameters are set to zero. The refinement of site n involves forming the tilted marginal at n and integrating out (u_n, f_n) to obtain analytically Z_n. Let the cavity distribution be $\mathcal{N}\left([u_n\ f_n]^T; \boldsymbol{\mu}^{\backslash n}, \mathbf{\Sigma}^{\backslash n}\right)$. We find that each component of

the noise model makes an independent contribution to Z_n, which we evaluate here for the standard model (3.5). Let $\lambda \in \{\pm 1\}$ select either real ($\lambda = 1$) or outlier ($\lambda = -1$), such that $Z_n = \sum_{\lambda\in\{\pm1\}} Z_n^{(\lambda)}$, where

$$\begin{aligned} Z_n^{(\lambda)} &= \iint \sigma(u_n)\mathcal{N}\big(y_n\,;\, f_n\,,\, \sigma_\lambda^2\big)\,\mathcal{N}\left(\left[\begin{array}{c} u_n \\ f_n \end{array}\right]\,;\, \boldsymbol{\mu}^{\backslash n}\,,\, \boldsymbol{\Sigma}^{\backslash n}\right)\mathrm{d}u\mathrm{d}f \\ &= \mathcal{N}\Big(y_n\,;\, \mu_f^{\backslash n}\,,\, \sigma_\lambda^2 + \Sigma_{ff}^{\backslash n}\Big)\,\sigma\left(\lambda\cdot\frac{\mu_u^{\backslash n} + \frac{\Sigma_{uf}^{\backslash n}}{\sigma_\lambda^2+\Sigma_{ff}^{\backslash n}}(y_n - \mu_f^{\backslash n})}{\sqrt{1+\Sigma_{uu}^{\backslash n} - \frac{(\Sigma_{uf}^{\backslash n})^2}{\sigma_\lambda^2+\Sigma_{ff}^{\backslash n}}}}\right), \end{aligned} \tag{3.6}$$

and the σ_λ^2 are the two sampling variances. By setting the site parameters appropriately we seek to match the posterior moments of the tilted distribution, which are revealed through derivatives of (3.6). These become rather involved, and to avoid the clutter of notation, they are presented, together with a derivation of the result above, in appendix C. Through (1.13), these derivatives provide the requisite values for $\mathbf{b}_n$ and $\boldsymbol{\Pi}_n$, after which the full posterior distribution can be updated with a rank-2 operation, in $\mathcal{O}\left((2N)^2\right)$. Visiting all N sites then costs $\mathcal{O}(4N^3)$, after which it is advisable to refresh the posterior from scratch (at cubic cost) by incorporating into the prior all current site functions (1.15), avoiding the loss of precision that creeps in after repeated low-rank updates. Additionally at this stage we can calculate the EP estimate for the marginal likelihood using (1.16), which provides a convenient indication of when convergence has occurred.[3] The entire process is summarised in algorithm 3. The cost of these EP iterations is cubic in the size of the covariance matrix, hence for the TGP time complexity is $\mathcal{O}(8N^3)$.

Model selection

We have seen how EP, in addition to the approximate moments of the posterior distribution, provides an estimate (1.17) of the derivatives of the evidence with respect to kernel hyperparameters. Since there are separate priors on the two processes, there are additional covariance parameters of $\mathbf{u}$ to optimize. It is also necessary to learn the base rate of corruption regulated by m_u, and the variances of the two noise models. For the latter, it is recommended to use logarithmic values $\log\sigma_O^2$ and $\log\sigma_R^2$ to allow

[3] Although the TGP provides no guarantee, it is usually found to settle down within a small number of iterations (five or so), and this number is typically independent of N, contributing only a constant factor to the algorithmic cost.

Algorithm 3 Estimating the posterior distribution of $[\mathbf{u}; \mathbf{f}]$ for the TGP

input: $\mathbf{m_u}$, $\mathbf{K_{uu}}$, $\mathbf{K_{ff}}$, $\mathbf{y}$
state: $\boldsymbol{\theta} = \left(\mathbf{h}; \mathbf{A}\right)$ the approximate posterior, $\left\{\mathbf{b}_n; \mathbf{\Pi}_n\right\}_{n=1}^{N}$ the site parameters
$\boldsymbol{\theta} := \left(\begin{bmatrix} \mathbf{m_u} \\ \mathbf{0} \end{bmatrix}; \begin{bmatrix} \mathbf{K_{uu}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K_{ff}} \end{bmatrix} \right)$ {initialize estimate to the prior}
while L not converged **do**
  for all $n \in \{1, 2, \ldots, N\}$ **do**
    obtain cavity parameters $\boldsymbol{\theta}^{\backslash n}$ by (1.11)
    calculate moments and derivatives by (1.12) and (C.1)–(C.4) as appropriate
    obtain site parameters $\mathbf{b}_n$, $\mathbf{\Pi}_n$ by (1.13)
    rank-2 update posterior $\boldsymbol{\theta}$ by (1.14)
  end for
  refresh $\boldsymbol{\theta}$ by (1.15)
  calculate the approximate log marginal likelihood L by (1.16)
end while
return: $\boldsymbol{\theta}$ (estimate of posterior), L (approximate log marginal likelihood)

for unconstrained optimization. Derivatives of the log marginal likelihood for the extra parameters are listed for completion in appendix C (for the kernels themselves, these are calculated in the same way as for standard GP regression models). These gradients can then be passed to an "off the shelf" optimizer to maximize the evidence, e.g. L-BFGS, a popular quasi-Newton method (Nocedal, 1980), or conjugate gradients (originally proposed in Hestenes and Stiefel (1952); Shewchuk (1994) provides a very lucid account, and Carl Rasmussen's Matlab implementation is available in his `gpml` package[4]).

Convergence

We find on occasion that the EP iterations fail to converge adequately. There are three measures we have employed to aid the process. First, the order of site refinement is randomized initally but then held constant throughout optimization (i.e. model selection); in this way, the posterior is less free to explore secondary peaks. A second modification retains the site parameters across calls to the EP subroutine (using Matlab's `persistent` tag), such that after the optimizer has made any modifications to the hyperparameters, EP resumes from a solution hopefully near where it left off. Not only does this encourage the same peak to be rediscovered, for which the gradient information returned to the optimizer will be consistent, but it speeds up the convergence of EP by initializing at a distribution close to the old posterior approximation. The

[4]Available from `http://www.gaussianprocess.org/gpml/`.

final alteration is where necessary to make *damped* updates: rather than change the site parameters to values for which the new posterior marginal matches the moments of the tilted distribution, we set them to a convex combination of new and old. This limitation can slow convergence, but helps prevent a sudden collapse of the posterior onto one peak, a problem we address in section 3.5.

3.2.4 Predictions

When we make predictions with the TGP, we must be precise about the nature of our query. Often, the data attributed to the "outlier" component consist only of nuisance noise to be eliminated, in which case we seek the distribution of the uncorrupted signal, whose variance is due solely to residual uncertainty from the prior. It is found by marginalizing over $\mathbf{u}$ in the posterior:

$$p(f_\star|\mathbf{x}_\star, \mathbf{X}, \mathbf{y}) = \mathcal{N}\big(f_\star\,;\, \mathbf{k}_{\star\mathbf{f}}\mathbf{K}_{\mathbf{ff}}^{-1}\mathbf{y}\,,\, k_{\star\star} - \mathbf{k}_{\star\mathbf{f}}\mathbf{K}_{\mathbf{ff}}^{-1}\mathbf{k}_{\mathbf{f}\star}\big)\,.$$

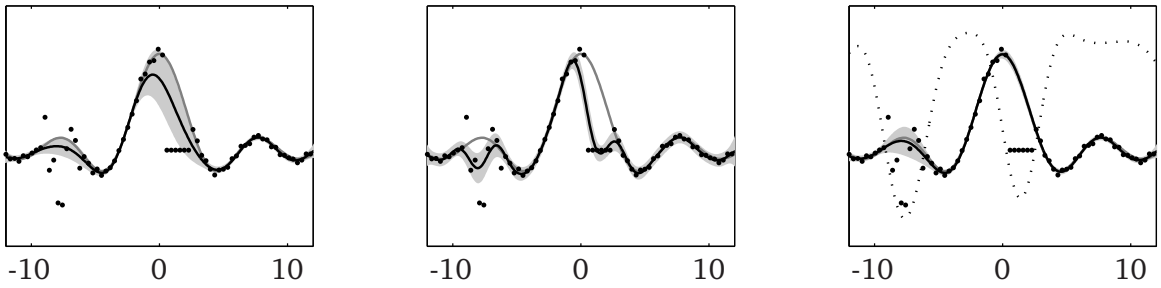
This is the marginal prediction of a full GP in itself, since we have not considered the mixture noise at all. However, for a general heteroscedastic signal, we will be interested in the distribution of *observations* at a test point, which requires consideration of the noise model and of the process on $\mathbf{u}$:

$$p(y_\star|\mathbf{x}_\star, \mathbf{X}, \mathbf{y}) = \iint p(y_\star|f_\star, u_\star)p(f_\star|\mathbf{x}_\star, \mathbf{X}, \mathbf{y})p(u_\star|\mathbf{x}_\star, \mathbf{X}, \mathbf{y})\mathrm{d}u_\star\mathrm{d}f_\star. \tag{3.7}$$

The final two terms of the integral are Gaussian, while the first is the TGP noise model, so that after marginalization we no longer recover a GP but the sum of two Gaussians (cf. (3.6)). By the same procedure used during inference, the moments of the distribution at $\mathbf{x}_\star$ can be evaluated analytically.

Empirically, we have found that the posterior *variance* on $\mathbf{u}$ may remain sufficiently large that error bars on the predictive distribution (3.7) are inappropriately wide.[5] In other words, the training data can allow reasonable estimates of the mixing proportions of the two components reflected in the mean of $\mathbf{u}$, but uncertainty in this estimate propagates as further uncertainty in $y_\star$. An expeditious solution is to fix $u_\star$ at

[5]This can occur particularly when the prior variance is driven to a large value in the model selection phase, a phenomenon we discuss in section 3.5.



(a) Mixture noise (b) Laplace noise (c) TGP

Figure 3.7: Three GP models with heavy tails. We illustrate two forms of corruption; in the first, around $x = -10$, data are scattered about the underlying mean. In the second, near $x = 0$, they appear in a tightly correlated cluster. The TGP is resilient to both, using the $\mathbf{u}$-process (dotted) to fit the mean (black) accurately. The mixture noise model tends to underfit, and is unable to cope with correlation in the outliers. Finally, a Laplace assumption appears in this case to be entirely inappropriate.

its posterior mean and integrate only over $f_\star$:

$$p(y_\star|\mathbf{x}_\star, \mathbf{X}, \mathbf{y}) \approx \int p(y_\star|f_\star, \mathbb{E}[u_\star])p(f_\star|\mathbf{x}_\star, \mathbf{X}, \mathbf{y})\mathrm{d}f_\star.$$

This simplification again yields a mixture of two Gaussians at the test point. The method will prove particularly valuable in chapter 4, where the noise model is extended to multiple $\mathbf{u}$-processes and moments of the exact marginal can no longer be calculated in closed form.

3.3 Experiments

There are two general noise characteristics for which the TGP may be well suited. The first occurs when the outlying observations appear in clusters; we have already seen how correlations in noisy targets can affect the standard mixture model, disrupting convergence, and hampering model selection because conflicting gradient information at the various local optima pull the hyperparameters in contrary directions. Fig. 3.7 shows a data set derived from the `sinc` function, and the inference of latent $\mathbf{f}$ by three heavy-tailed models.[6] We introduce two modes of corruption to the data, the first correlated only in the input domain but otherwise widely and symmetrically spread around the latent function, and the second additionally correlating the noisy targets.

[6]Since EP occasionally fails to converge for mixture noise (3.1), we provide here and in tests below the results for a Laplace model—again trained by EP, for which convergence is very reliable.

The mixture model seems to be a safe assumption for the scattered noise, since without consistency between the observations there is no substantial posterior peak associating any of these data closely with the latent function. Note however that some underfitting has occurred: the inferred distribution is rather smoother than the true `sinc` function. Where there are correlations in the corruption, the fixed weight apportioned by the model to the outlier component is clearly inappropriate, pulling the posterior mean away from the underlying function and leading to broader error bars in the vicinity of the outliers. To obtain this solution, we ran EP for a range of initializations and randomized the order of site refinement, displaying only the result most favoured by the evidence. In the case of the Laplace model, the learned lengthscale is rather too short and consequently it overfits in the two noisy regions. Meanwhile, the TGP exhibits resilience to both forms of corruption; furthermore, the solution is stable, such that hyperparameters can be learned reliably for a range of initializations. Depending on the application, the secondary $\mathbf{u}$ process may also contain useful information such as relative measures of confidence in the various observations.

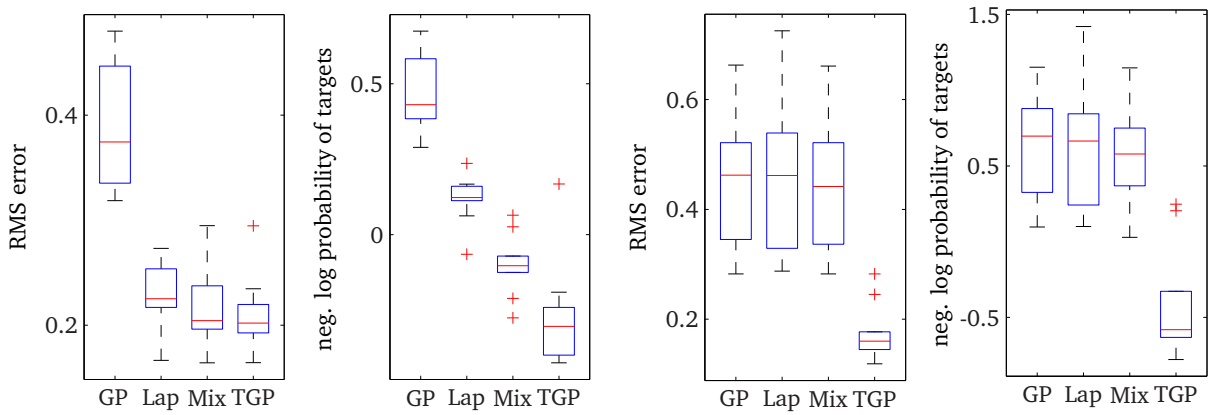
Friedman data

The `sinc` data illustrate well the concepts and mechanism of the TGP, differentiating it from other heavy-tailed models. However, the noise distribution was chosen adversarially, so we consider also data on which classical robust methods do perform well: these are a variation on a set of Friedman (1991), which appeared subsequently in Kuss (2006). The samples are drawn from a function of ten-dimensional vectors $\mathbf{x}$ which depends only on the first five components:

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

We generated ten sets of 90 training examples and 10000 test examples by sampling $\mathbf{x}$ uniformly in $[0, 1]^{10}$, and adding to the training data noise distributed $\mathcal{N}(0, 1)$. In our first experiment, we replicated the procedure of Kuss: ten training points were added at random with outputs sampled from $\mathcal{N}(15, 9)$ (a value likely to lie in the same range as f). The results appear as Friedman (1) in fig. 3.8. Observe that the r.m.s. error for all the robust methods is similar, but the TGP can model the variance more accurately than other GPs, by shrinking the predictive variance in regions without outliers.

In Friedman (2), the training set was augmented with two Gaussian clusters each of five noisy observations. The cluster centres were drawn uniformly in $[0, 1]^{10}$, with vari-



(a) Friedman (1) (b) Friedman (2)

Figure 3.8: Box-whisker plots for results on the Friedman set for the standard GP (i.i.d. Gaussian noise), Laplace, mixture and TGP models.

ance fixed at 10^{-3}. Output values were then drawn from $\mathcal{N}(0,1)$ for all ten points, to give highly clustered and correlated outliers distant from the underlying function. Now the TGP excels where the other methods are little improvement on the non-robust GP; it also yields very confident predictions (cf. Friedman (1)), because once the outliers have been accounted for there are fewer corrupted regions. In both experiments, the training data were renormalized to zero mean and unit variance, and throughout, we used the anisotropic squared exponential for the $\mathbf{f}$ process for automatic relevance determination, and an isotropic version for $\mathbf{u}$. The approximate marginal likelihood was maximized on three to five randomly initialized models; we chose for testing the most favoured.

3.3.1 Heteroscedastic noise

In the foregoing examples, inference has been of the latent $\mathbf{f}$. In the second domain of application, we consider data whose "noise" component is more regular and input-dependent, for which we seek directly to model the observations $\mathbf{y}$. A toy example is provided, akin to that appearing in Goldberg et al. (1998), where data are drawn from the `cos` function, and corrupted with noise whose standard deviation varies sinusoidally at lower frequency across the domain. Fig. 3.9a illustrates the predictive solution of the TGP, in which squared exponential kernels were used for both processes, and all hyperparameters were optimized by gradient ascent on the evidence.

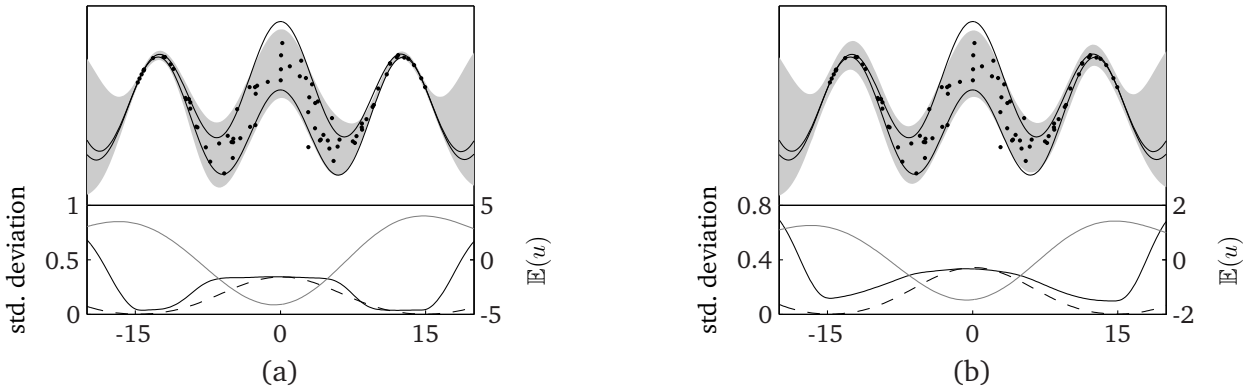


Figure 3.9: TGP inference for data with smoothly varying heteroscedastic noise. The grey region and lines mark a 95% confidence interval, the former of the posterior on $\mathbf{y}$ inferred by the TGP, the latter for the underlying process. In the lower plots appear the corresponding standard deviations (left axis; solid black for the TGP, dashed for the true function), and the mean of the $\mathbf{u}$-process (right axis; grey).

The model is successful in that it has learned to differentiate between the large variance noise around $x = 0$, and the very low variance nearer $x = \pm 15$. However, there are two shortcomings of the solution, and we find that these are characteristic of the model. First, because in reality the process noise is not considered directly (rather, it is simulated via the Gaussian mixture), it suffers saturation in the variance predictions: when all the weight has been apportioned by $\mathbf{u}$ to one or other component, the predictive variance can be pushed no further; these regions appear as flat plateaus or valleys in the lower plot of panel (a). Fortunately in this case, the TGP has learned values for σ_R^2 and σ_O^2 which allow it to mimic the true noise process comparatively well.

However, a second problem appears here: the true error on observations varies quite gently, whereas the predicted standard deviation makes a relatively sharp transition from the outer regions, where the model is confident data have been generated by the "real" component, to the inner, where it is all but certain the data are "outliers". This is perhaps surprising, since in qualitative terms, the process on $\mathbf{u}$ appears to vary with the desired frequency (that of the true error process). In fact, the issue is caused by the nonlinear transformation of $\mathbf{u}$ made by the probit, and by the influence on the mixture variance exerted by the variance of the outlier distribution. This latter effect explains why the ML-II solution demands such a large magnitude for the $\mathbf{u}$-process (which has a range ± 5): in fig. 3.9b is illustrated the effect of rescaling its posterior mean. Although we can better approximate the gradual change in noise variance, we suffer a paranoiac effect in regions where before the model enjoyed great confidence.

The problem is, since the "outlier" component has a relatively large variance (in the example, two orders of magnitude larger than that of the "real"), it needs only a small fraction of the total weight to have a disproportionately influential effect on the variance of the mixture: by increasing the range over which a transition between the two components occurs, we have prevented the model from expressing absolute certainty about either. In general, when we make predictions of $y_\star$ rather than $f_\star$, the larger variance component tends to dominate the prediction for all but large positive values of u. If we wish to avoid this pollution in regions of confidence, we require a large amplitude for the kernel, but in consequence, regions of intermediate variance may be modelled inaccurately when $\mathbf{u}$ sweeps precipitately through the zone of sensitivity.

A partial solution to these difficulties may be provided by adding extra outlier processes to cater for a range of possible deviations; this approach is explored in chapter 4. A more exotic alternative is to "warp" the process on $\mathbf{u}$, as described by Snelson et al. (2004). By learning a nonlinear transformation of observations with non-Gaussian noise, data in the latent space are rescaled in a supervised manner to be well-modelled by a GP. The warping function is constrained to be monotonic, allowing inference in the observation domain via an inverse operation.[7] Applied to the TGP, the warping function could be learned simultaneously with the optimization of model parameters, or as a post-processing step once the distribution of $\mathbf{u}$ has been established. We defer development of these ideas to future work.

Motorcycle data

As a final example, we consider the behaviour of the TGP on the one-dimensional *motorcycle* set (Silverman, 1985), which is strongly heteroscedastic. The original data are spread widely, and for the reasons discussed above we found it helpful to renormalize their output to unit variance. We see in fig. 3.10a that all GP methods model the mean of the process equally well, but the TGP is able to provide a better fit to the variance (the difference is not as marked as the Friedman examples partially because the set is very small: with only 133 points, we created twelve folds of the data, holding out for testing each time a different set of eleven points (one had twelve)). For these data, we found the EP updates required heavy damping in order to achieve adequate convergence, and even with this precaution certain hyperparameter settings caused insoluble difficulties for the optimizer; our results were obtained from initializing several

[7] Note that the warped GP is useful when the magnitude of corruption depends on the *output* value, in constrast to the TGP, which is an input-dependent model.

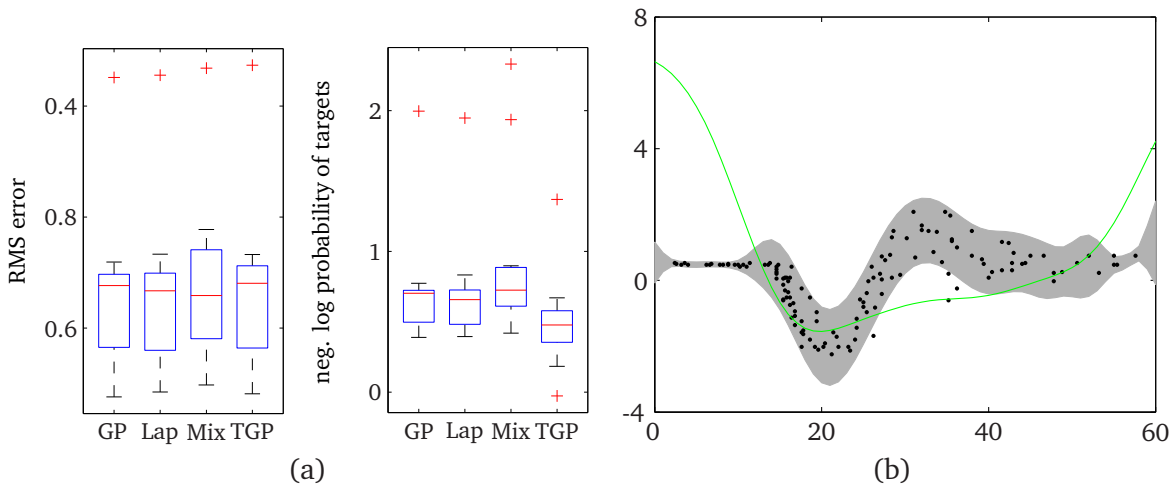


Figure 3.10: In panel (a) we compare the performance of four GP models on Silverman's motorcycle data set. Panel (b) illustrates the posterior obtained for one run of the TGP, where the green line shows the mean posterior on $\mathbf{u}$.

runs at random hyperparameter settings and testing the model with greatest estimated marginal likelihood. We discuss the results in more detail in the following section, where comparisons are made with the posterior obtained by Monte Carlo methods.

3.4 Stochastic inference

Although our implementation has proven useful empirically, the faithfulness of the posterior approximation obtained by EP to that of the underlying TGP model remains unclear. In this section, a Markov chain Monte Carlo (MCMC) method is described which attempts to draw independent samples from the true posterior, allowing comparisons between true and approximate solutions. From the perspective of MCMC, the TGP model is related both to the mixture of Gaussians (3.1) used often in robust regression, and to the conventional model for GP classification: to the former because, by introducing a set of latent variables $c_n \in \{\pm 1\}$ that assigns observations to either the real $(+1)$ or the outlier (-1) model, the likelihood conditional on $\mathbf{c}$ becomes Gaussian: let

$$s_{nn} = \begin{cases} \sigma_R^2 & \text{if } c_n = 1, \\ \sigma_O^2 & \text{if } c_n = -1, \end{cases} \qquad \text{where} \qquad \mathbf{S} = \mathsf{diag}(\mathbf{s}), \tag{3.8}$$

and $p(c_n|u_n) = \sigma(c_n u_n)$. Then

$$p(\mathbf{y}|\mathbf{c}) = \int p(\mathbf{y}|\mathbf{f},\mathbf{c})p(\mathbf{f}|\mathbf{X})\mathrm{d}\mathbf{f} = \mathcal{N}(\mathbf{y}\,;\,\mathbf{0}\,,\,\mathbf{K_{ff}}+\mathbf{S})\,.$$

This property allows us to avoid sampling explicitly the $\mathbf{f}$: they are always marginalized. We can then use Gibbs sampling for each c_n.

The mixing weights vary in the TGP as a sigmoidal function of u_n, making marginalization of $\mathbf{u}$ intractable, and requiring explicit samples of the $\mathbf{u}$-process. This reveals an isomorphism to a GP classification model in which the target labels are $\mathbf{c}$. There are two approaches to sampling from the posterior in the literature: Neal (1997) uses Gibbs sampling to update sequentially the components of the latent vector, while Kuss and Rasmussen (2005) use "hybrid" or Hamiltonian Monte Carlo (HMC). In the following, the latter method is adopted since HMC will in any case be used to sample the model hyperparameters.

3.4.1 Inference

The Monte Carlo chain proceeds in two stages, broadly alternating between $\mathbf{c}|\mathbf{u}$ and $\mathbf{u}|\mathbf{c}$. In the first, the $\mathbf{c}$ are updated by Gibbs sampling from the N Bernoulli distributions

$$p(c_n|\mathbf{c}^{\setminus n},\mathbf{u},\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{c})p(c_n|\mathbf{c}^{\setminus n},\mathbf{u})}{p(\mathbf{y}|\mathbf{c}^{\setminus n},\mathbf{u})} = \frac{p(\mathbf{y}|\mathbf{c})p(\mathbf{c}|\mathbf{u})/p(\mathbf{c}^{\setminus n}|\mathbf{u})}{p(\mathbf{y},\mathbf{c}^{\setminus n}|\mathbf{u})/p(\mathbf{c}^{\setminus n}|\mathbf{u})} = \frac{p(\mathbf{y}|\mathbf{c})p(\mathbf{c}|\mathbf{u})}{p(\mathbf{y},\mathbf{c}^{\setminus n}|\mathbf{u})};$$

discarding terms independent of c_n leaves only the weighted evidence expressions

$$\pi_R = p(\mathbf{y}|\mathbf{c}^{\setminus n}, c_n = 1)\sigma(u_n) \qquad \text{and} \qquad \pi_O = p(\mathbf{y}|\mathbf{c}^{\setminus n}, c_n = -1)\sigma(-u_n),$$

hence

$$p(c_n = 1|\mathbf{c}^{\setminus n},\mathbf{u},\mathbf{y}) = \frac{\pi_R}{\pi_R+\pi_O} \qquad \text{and} \qquad p(c_n = -1|\mathbf{c}^{\setminus n},\mathbf{u},\mathbf{y}) = \frac{\pi_O}{\pi_R+\pi_O}. \tag{3.9}$$

The log evidence term itself is

$$\log p(\mathbf{y}|\mathbf{c}) = -\frac{1}{2}\mathbf{y}^T\left(\mathbf{K_{ff}}+\mathbf{S}\right)^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K_{ff}}+\mathbf{S}| - \frac{N}{2}\log(2\pi). \tag{3.10}$$

Calculating (3.10) for $c_n \in \{\pm 1\}$ would appear prohibitively expensive, but the complexity at each iteration is reduced to $\mathcal{O}(N^2)$ if a low rank update to (the Cholesky

decomposition of) $\mathbf{K_{ff}} + \mathbf{S}$ is made. The updated covariance is retained in case the proposal is accepted, else the original is restored.

Monte Carlo methods for mixture models are often troubled by the inability of Gibbs sampling to achieve coordinated updates: traversing the various modes of the posterior that correspond to different assignments of $\mathbf{c}$ usually requires the simultaneous adjustment of several c_n because intermediate states have low probability. To help overcome the poor mixing of the Markov chain, *overrelaxation* is employed, specifically the ordered overrelaxation of Neal (1995) which generalizes the ideas of Adler (1981) to non-Gaussian distributions. The intuition is that we encourage c_n to change as frequently as possible while maintaining the correct marginals: let π_n be the probability of switching the component assignment $c_n := -c_n$ as determined by (3.9). Using ordered overrelaxation, we flip the component assignment with the revised probability

$$\hat{\pi}_n = \mathsf{min}\left(1, \frac{\pi_n}{1-\pi_n}\right).$$

A simple derivation of this result is presented in appendix C.

In the second stage of the chain, all remaining parameters, including hyperparameters of the kernel, are updated using Hamiltonian Monte Carlo (see section 1.5). This requires expressions for the negative log probability and its derivatives. The joint is

$$\begin{aligned} p(\mathbf{u}, \sigma_R^2, \sigma_O^2, m_\mathbf{u}, \boldsymbol{\psi}_\mathbf{u}, \boldsymbol{\psi}_\mathbf{f} | \mathbf{c}, \mathbf{X}, \mathbf{y}, \boldsymbol{\zeta}) &\propto p(\mathbf{y}|\mathbf{c}, \boldsymbol{\psi}_\mathbf{f}, \mathbf{X}) p(\mathbf{c}|\mathbf{u}) p(\mathbf{u}|\mathbf{X}, m_\mathbf{u}, \boldsymbol{\psi}_\mathbf{u}) p(\sigma_R^2, \sigma_O^2, m_\mathbf{u}, \boldsymbol{\psi}_\mathbf{u}, \boldsymbol{\psi}_\mathbf{f}|\boldsymbol{\zeta}) \\ &= \mathcal{N}(\mathbf{y}\,;\,\mathbf{0}\,,\,\mathbf{K_{ff}} + \mathbf{S}) \prod_{n=1}^{N} \sigma(c_n u_n) \mathcal{N}(\mathbf{u}\,;\,\mathbf{m_u}\,,\,\mathbf{K_{uu}})\, p_0, \end{aligned}$$

where $p_0 = p(\sigma_R^2, \sigma_O^2, m_\mathbf{u}, \boldsymbol{\psi}_\mathbf{u}, \boldsymbol{\psi}_\mathbf{f}|\boldsymbol{\zeta})$ allows non-uniform priors on the model parameters and hyperparameters, and $\mathbf{m_u} = m_\mathbf{u}\mathbf{1}$. Within a constant term c,

$$\begin{aligned} -\log p(\mathbf{u}, \sigma_R^2, \sigma_O^2, m_\mathbf{u}, \boldsymbol{\psi}_\mathbf{u}, \boldsymbol{\psi}_\mathbf{f} | \mathbf{c}, \mathbf{X}, \mathbf{y}, \boldsymbol{\zeta}) &= \frac{1}{2}\log|\mathbf{K_{ff}} + \mathbf{S}| + \frac{1}{2}\mathbf{y}^T(\mathbf{K_{ff}} + \mathbf{S})^{-1}\mathbf{y} \\ &\quad - \sum_{n=1}^{N} \log\sigma(c_n u_n) + \frac{1}{2}\log|\mathbf{K_{uu}}| + \frac{1}{2}(\mathbf{u} - \mathbf{m_u})^T \mathbf{K_{uu}}^{-1}(\mathbf{u} - \mathbf{m_u}) + \log p_0 + \mathsf{c}. \end{aligned}$$

There is a marked difference in complexity for different updates: a change in $\mathbf{u}$ or $m_\mathbf{u}$ costs only $\mathcal{O}(N^2)$; adjusting the noise parameters σ_R^2 and σ_O^2 or varying the kernel

hyperparameters requires the recalculation of a full $N \times N$ matrix inverse. For speed, it is therefore sensible to make these latter updates only comparatively infrequently.

The Markov chain may be initialized by using the ML parameters obtained after running the EP method, or more simply by a preprocessing stage of standard GP regression, after which we may guess $\sigma_O^2 := 2\sigma_R^2$. Hyperparameters of the kernel for the $\mathbf{u}$ process are set equal to those of the $\mathbf{f}$ process, and its mean is initialized to zero or a small positive value.

3.4.2 Prediction

Given a set of posterior samples $\left[\mathbf{u}^{(t)}, \mathbf{c}^{(t)}, \sigma_R^{2\,(t)}, \sigma_O^{2\,(t)}, m_{\mathbf{u}}^{(t)}, \boldsymbol{\psi}_{\mathbf{u}}^{(t)}, \boldsymbol{\psi}_{\mathbf{f}}^{(t)}\right]_{t=1}^{T}$, predictions of $f_\star$ are given by

$$\begin{aligned} p(f_\star|\mathbf{x}_\star, \mathbf{X}, \mathbf{y}) &= \int p(f_\star|\mathbf{f}) p(\mathbf{f}|\mathbf{X}, \mathbf{y}) \mathrm{df} \\ &\approx \frac{1}{T}\sum_{t=1}^{T} \int p(f_\star|\mathbf{f}) \frac{p\left(\mathbf{y}\middle|\mathbf{f}, \mathbf{c}^{(t)}, \sigma_R^{2\,(t)}, \sigma_O^{2\,(t)}\right) p\left(\mathbf{f}\middle|\mathbf{X}, \boldsymbol{\psi}_{\mathbf{f}}^{(t)}\right)}{p\left(\mathbf{y}\middle|\mathbf{c}^{(t)}, \sigma_R^{2\,(t)}, \sigma_O^{2\,(t)}\right)} \mathrm{df} \\ &= \frac{1}{T}\sum_{t=1}^{T} \mathcal{N}\left(f_\star\,;\, \mu_f^{(t)}\,,\, \sigma_f^{2(t)}\right), \end{aligned}$$

where

$$\mu_f^{(t)} = \mathbf{K}_{\mathbf{f}\star}^{(t)} \left(\mathbf{K}_{\mathbf{ff}}^{(t)} + \mathbf{S}^{(t)}\right)^{-1} \mathbf{y}; \qquad \sigma_f^{2(t)} = \mathbf{K}_{\star\star}^{(t)} - \mathbf{K}_{\star\mathbf{f}}^{(t)} \left(\mathbf{K}_{\mathbf{ff}}^{(t)} + \mathbf{S}^{(t)}\right)^{-1} \mathbf{K}_{\mathbf{f}\star}^{(t)},$$

and $\mathbf{S}^{(t)}$ depends on $\mathbf{c}^{(t)}, \sigma_R^{2\,(t)}$ and $\sigma_O^{2\,(t)}$, and the various $\mathbf{K}_{\cdot\cdot}^{(t)}$ all depend on $\boldsymbol{\psi}_{\mathbf{f}}^{(t)}$. If predictions of $y_\star$ are required, the distribution of $u_\star$ is also important:

$$\begin{aligned} p(y_\star|\mathbf{x}_\star, \mathbf{X}, \mathbf{y}) &= \iint p(y_\star|u_\star, f_\star) \iint p(u_\star, f_\star|\mathbf{u}, \mathbf{f}) p(\mathbf{u}, \mathbf{f}|\mathbf{X}, \mathbf{y}) \mathbf{du df} \mathrm{d}u_\star \mathrm{d}f_\star \\ &\approx \frac{1}{T}\sum_{t=1}^{T} \iint p(y_\star|u_\star, f_\star) \mathcal{N}\left(u_\star\,;\, \mu_u^{(t)}\,,\, \sigma_u^{2(t)}\right) \mathcal{N}\left(f_\star\,;\, \mu_f^{(t)}\,,\, \sigma_f^{2(t)}\right) \mathrm{d}u_\star \mathrm{d}f_\star \\ &= \frac{1}{T}\sum_{t=1}^{T} \sum_{\lambda\in\{\pm 1\}} \left\{ \sigma\left(\lambda \cdot \frac{\mu_u^{(t)}}{\sqrt{1+\sigma_u^{2(t)}}}\right) \mathcal{N}\left(y_\star\,;\, \mu_f^{(t)}\,,\, \sigma_f^{2(t)} + \sigma_\lambda^{2(t)}\right) \right\}, \end{aligned}$$

where λ has been reintroduced to keep the equation concise and carries the same meaning as (3.6), and where

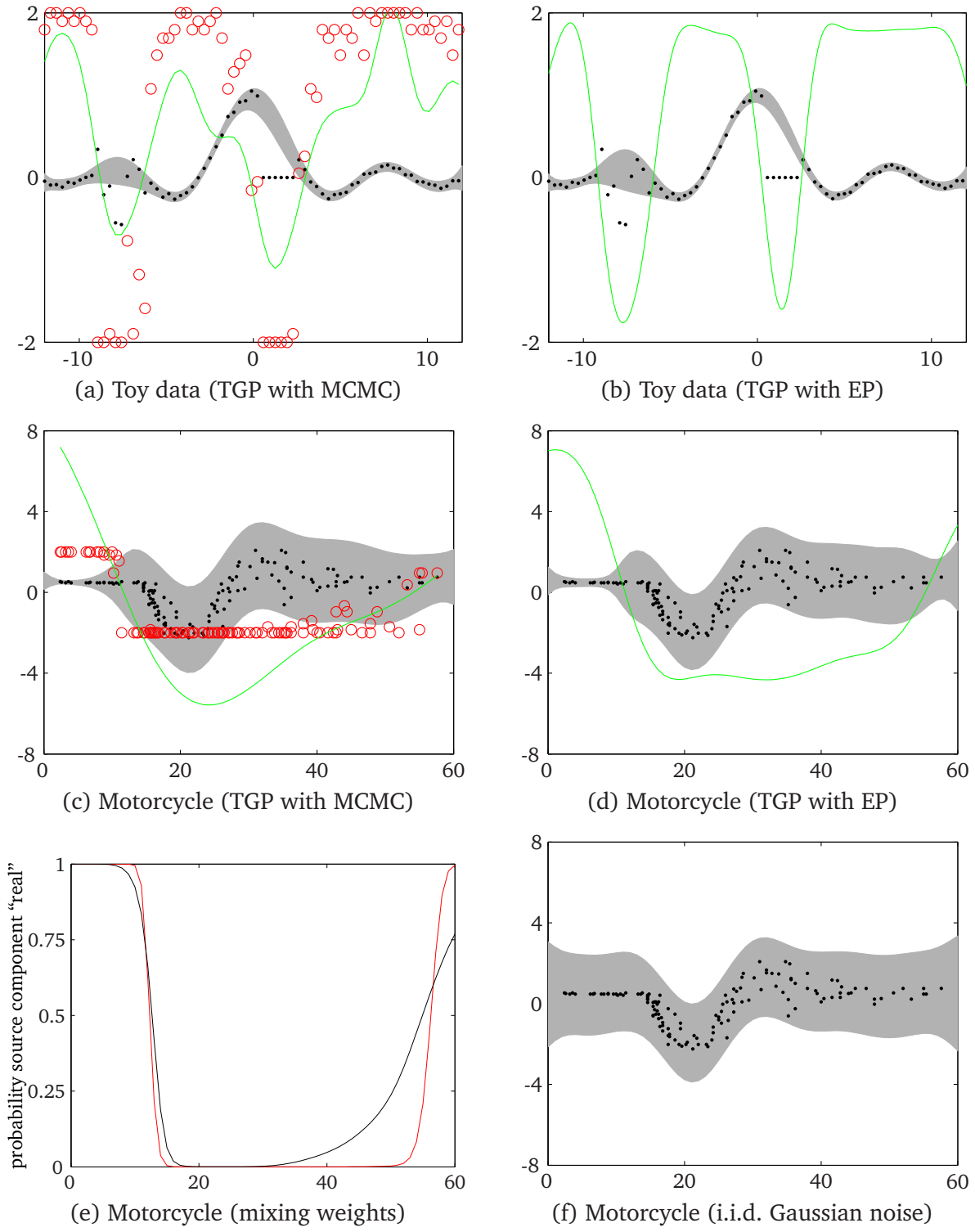
$$\mu_u^{(t)} = m_{\mathbf{u}}^{(t)} + \mathbf{k}_{\mathbf{u}\star}^{(t)T} \left(\mathbf{K}_{\mathbf{uu}}^{(t)}\right)^{-1} \left(\mathbf{u}^{(t)} - \mathbf{m}_{\mathbf{u}}^{(t)}\right); \qquad \sigma_u^{2\,(t)} = k_{\star\star}^{(t)} - \mathbf{k}_{\mathbf{u}\star}^{(t)T} \left(\mathbf{K}_{\mathbf{uu}}^{(t)}\right)^{-1} \mathbf{k}_{\mathbf{u}\star}^{(t)}.$$

3.4.3 Experiments

We explore the posterior for the two kinds of noise distribution that concerned us earlier: data with clusters of outliers, and data with more smoothly heteroscedastic noise. In the first case, refer to fig. 3.11a. The mode around which the Monte Carlo algorithm has sampled corresponds fairly closely with that fit by EP (fig. 3.11b). The variance on the latent $\mathbf{f}$ is a good match for that located in the MCMC iterations, although some subtleties of the $\mathbf{u}$ process have been lost. The relatively small magnitude for the $\mathbf{u}$ process in this case has also allowed considerable uncertainty in the class assignment for data near the corrupted regions.

If we consider now the remaining panels of fig. 3.11 we see a different side to the TGP behaviour in fitting heteroscedastic data. The solution obtained by MCMC is in fact rather poor: the variance estimate switches quite abruptly from a narrow band around the data between $t = 0$ to $t = 10$, to a much broader distribution further along the t axis. This aspect of the posterior is captured well by the EP fit, although again we observe some loss of smoothness in the estimate of $\mathbf{u}$, observed clearly in fig. 3.11e: in black is the probability of the input being labelled "real" according to the MCMC model, while in red is the EP estimate. The latter behaves essentially like a binary switch, which is what we observe in fig. 3.11d.

At first glance this is rather disappointing. We explain the problem as follows: in fig. 3.11f is the posterior obtained for a GP using an i.i.d. Gaussian noise assumption. Except for its inability to shrink its variance estimate in the region on the left identified earlier, the distribution is in fact very similar to that obtained by the TGP under both MCMC and EP. What our model has done is break the task into essentially two regions and learned a GP to fit each, approximately independently. (We will consider harnessing this "mixture of experts" in section 4.4.) Here then is a case of model mismatch, and that perspective is corroborated by Rasmussen and Ghahramani (2002, sec. 5), where an infinite mixture of GPs is sampled from by MCMC: "the posterior distribution of number of needed GPs has a broad peak between 3 and 10, where less than 3 occupied experts is very unlikely, and above 10 becoming progressively less likely".



(a) Toy data (TGP with MCMC)

(b) Toy data (TGP with EP)

(c) Motorcycle (TGP with MCMC)

(d) Motorcycle (TGP with EP)

(e) Motorcycle (mixing weights)

(f) Motorcycle (i.i.d. Gaussian noise)

Figure 3.11: Data appear as black dots. Two standard deviations around the inferred mean are shown in grey; for panels (a) and (b) we consider variance on the latent $\mathbf{f}$, while the remaining panels consider that on observed $\mathbf{y}$. The green line indicates the mean of the latent $\mathbf{u}$ process, while the red circles in panels (a) and (c) indicate how frequently the associated data were assigned to the "real" noise model (red circles at +2 means "always") or the "outlier" model (red circles at -2 means "always") by the sampling routine.

3.5 Discussion

The generality of GPs means they can easily be allied with any non-Gaussian noise model to provide a more robust regressor (although the convenience of a tractable inference procedure is usually lost). Choices which have been applied widely, and which are reviewed in detail by Kuss (2006), were discussed above: the mixture of two Gaussians, the double exponential, and Student's t distribution. Since the essential structure of the GP remains unchanged, in all these cases the noise is assumed i.i.d., and none may be appropriate when errors appear with their own structure.

We are not aware of any work that explicitly targets the problem of clustered outliers. However, several solutions to problems of more general heteroscedasticity that build on a GP framework have been proposed. One of the earliest, due to Goldberg et al. (1998), is similar in design to the twinned GP, but the second process is placed on the log variance itself. Inference is analytically intractable so Gibbs sampling is used to generate noise vectors from the posterior distribution by alternately fitting the signal process and fitting the noise process. A further stage of sampling is required at each test point to estimate the predictive variance, and model hyperparameters are sampled by Metropolis-Hastings (although more efficient Hamiltonian methods would be applicable). As we have suggested, the TGP has a similar flavour but sacrifices the flexibility of directly fitting the variance for a more efficient inference based on EP. We envisage slightly different domains for the two models: if an accurate estimate of variance is required in a truly heteroscedastic domain, the approach of Goldberg et al. is certainly superior. If the data are polluted with outliers, or accuracy of the variance prediction is less important than speed of inference, the TGP would seem more suitable.

Two papers which address models involving mixtures of GPs for heteroscedastic modelling are Tresp (2000) and Rasmussen and Ghahramani (2002). The former constructs a mixture of a prespecified number M of GPs by fitting three sets of supplementary GPs over the means, variances, and the gating process; each of these uses M processes. Tresp appears to assume knowledge of the correct hyperparameters, after which inference is by an EM-style procedure, although a more general Monte Carlo inference would certainly be feasible. The latter paper is more adventurous, proposing a potentially infinite mixture, for which the correct number of components is determined as part of the inference. This is achieved by deriving a localized estimate of "occupation number" for use in a Dirichlet process: in each sweep, each datum may be assigned either to an existing component or to a fresh (unpopulated) component. This

strict partition of the inputs bestows computational benefits (each GP models only a subset of the data, and an upper bound on its size can be enforced by adjusting the component assignment probabilities), as well as preventing the contamination of predictions with data fit by other components; these advantages are not present in the model of Tresp, and indeed nor in the TGP since with a deterministic approximation it is hard to see how such partitioning could be achieved. Hence, for Tresp, time complexity is $\mathcal{O}(3MN^3)$, while for Rasmussen and Ghahramani each iteration of the Monte Carlo algorithm is $\mathcal{O}(N^3/M)$ provided the data have been divided equally amongst the experts.

Cawley et al. (2003) and Le et al. (2005) take the regularization view of obtaining a MAP predictor of the GP mean, treating the inference procedure directly as an optimization process. In both cases are suggested efficient procedures for deriving "unbiased estimates" of the input-dependent variance, but their regularization perspective precludes a principled approach to learning the kernel parameters, both employing cross-validation.

The ingenious suggestion of Kersting et al. (2007) requires no additional machinery beyond standard homoscedastic GP regression, and will also fit parameters of the kernel. The objective is calculation of the "most likely" variances: first, a standard GP is fit to the data, and from its predictions we derive empirical estimates of the variance z_n at each observation. A second data set is created, with ordinates $\mathbf{X}$ and targets $\mathbf{z}$—the estimated variances—on which we learn another GP model. In conjunction with the first, this yields a third, essentially heteroscedastic, GP but for which inference is tractable: the known variances are simply added to the diagonal of the full covariance matrix. These last two steps are repeated until convergence, although the authors note this is not guaranteed and may be to an inferior local optimum. A comparison of their model with ours, and also with that of Goldberg et al. (1998), would form an interesting avenue for future research. In particular, we are curious how robust the "most likely" regressor would be towards clustered outliers.

The SPGP model for sparse GP regression (Snelson and Ghahramani, 2006a), described in chapter 2 in a more general setting, embodies a non-stationary kernel function parameterized by pseudo-inputs. By taking advantage of the "pinching" effect that reduces the variance in the region of these points, it is possible to fit heteroscedastic data better than a standard i.i.d. Gaussian GP. Snelson and Ghahramani (2006b) extend the idea further by associating with each pseudo-input a weight which allows it

gradually to be "turned off", broadening the variance around that point. Empirically they observe some benefits with the extension, but learning the additional parameters can lead to overfitting on some small data sets.

3.5.1 Convergence

Although not flawless, we have found better convergence when EP is run on the TGP than when used with a conventional mixture model, and particularly on data with clustered outliers (for more widely heteroscedastic data, the TGP often requires heavy damping to allow convergence). In this section, we explore the modes of failure of the mixture, and explain how the TGP ameliorates these problems with its extra process on $\mathbf{u}$. Initially, however, we consider how EP behaves for log concave likelihoods in general, since it is conjectured (Rasmussen and Williams, 2006, sec. 3.6) but unproven that it will always converge in such cases, and they provide an instructive example as to how relaxing the constraint of log concavity can introduce difficulties.

An intuitive understanding of the stability of EP comes from appreciating how each site refinement modifies the covariance. First, it is evident from the definition of concave functions that they preserve the log concavity of the marginal prior: if $p(x)$ and $q(x)$ are log concave, i.e. if for all $\alpha \in [0, 1]$, and for all $x, x' \in \mathbb{R}$,

$$\begin{aligned} & p(\alpha x + (1-\alpha)x') \geq p(x)^{\alpha} p(x')^{(1-\alpha)} \quad \text{and similarly for } q(\cdot), \\ \text{then} \quad & p(\alpha x + (1-\alpha)x') q(\alpha x + (1-\alpha)x') \geq \left(p(x)q(x)\right)^{\alpha} \left(p(x')q(x')\right)^{(1-\alpha)}, \end{aligned}$$

and the product is log concave, thus guaranteed to remain unimodal. We can make a stronger statement: recall that, for a Gaussian cavity distribution and log concave likelihood $q(f)$, the normalizing constant of the tilted distribution is

$$Z = \int q(f) \mathcal{N}\left(f \,;\, \mu \,,\, \sigma^2\right) \mathrm{d}f;$$

first, as a function of μ, it can be shown Z is also log concave by virtue of this property in its components (Bogachev, 1998, sec. 1.8). Recall further that

$$\frac{\partial Z}{\partial \mu} = \frac{Z}{\sigma^2}\big(\mathbb{E}[f] - \mu\big) \qquad \text{and} \qquad \frac{\partial^2 Z}{\partial \mu^2} = \frac{Z}{\sigma^4}\left(\mathbb{V}[f] - \sigma^2 + \big(\mathbb{E}[f] - \mu\big)^2\right),$$

$$\text{hence} \qquad \frac{\mathbb{V}[f] - \sigma^2}{\sigma^4} = \frac{1}{Z}\frac{\partial^2 Z}{\partial \mu^2} - \frac{1}{Z^2}\left(\frac{\partial Z}{\partial \mu}\right)^2,$$

which is an expression for the change in marginal variance, and is non-positive if the variance of the tilted distribution is bounded by that of the cavity distribution. Indeed, since Z is log concave, it can be shown (Bergstrom and Bagnoli, 2005, lemma 4) that $\frac{1}{Z^2}\left(\frac{\partial Z}{\partial \mu}\right)^2 \geq \frac{1}{Z}\frac{\partial^2 Z}{\partial \mu^2}$; we conclude the variance never grows beyond the cavity, and in consequence the EP iterations are "well-behaved". This is an intentionally vague statement, since it remains possible for changes at other sites to adjust the posterior in such a way that, although the site refinement gives a marginal variance no larger than the cavity, it may still be larger than the variance had been before refinement. It is this feature of the algorithm that contributes to making a convergence proof so elusive.

Consider now the standard heavy-tailed likelihoods: except for the Laplace, these are not log concave (refer to fig. 3.3), and in general we can make no guarantees about the variance of the updated marginal. Particularly when the two modes of the tilted distribution (corresponding to explanations of the observation as genuine or erroneous) are of comparable magnitude, its variance may become greater than that of the cavity. This plagues convergence of EP-style algorithms: the iterations can initially settle on a certain mode only for subsequent site refinements to demand a revised interpretation. In consequence it becomes hard to determine if or when the algorithm has found a satisfactory solution.

Although we have seen that EP encourages the approximation to place mass everywhere the tilted distribution does (by which means we hope to account for both modes of the marginal), one peak usually dwarfs the other, sometimes so dominating the moment contributions that the approximation essentially disregards altogether the alternative interpretation. Usually this is desirable, but if an outlier has mistakenly been classified "real", the assignment can be difficult to reverse because the posterior is pulled sharply towards the observation. Rather, without substantial evidence of their legitimacy, nearby observations tend to be regarded instead as outliers: the preference of the heavy tails is to conform to the current hypothesis. In this respect, EP can be very sensitive to the order in which sites are accumulated in its initial "assumed density filtering" loop, and we will see this problem is more severe with the mixture of Gaussians noise model than with that of the TGP.

A further difficulty can arise, particularly when we revisit an outlier mislabelled as real and attempt to reform the cavity distribution: having incorporated the other sites, the marginal variance σ_n^2 at the outlier may have grown, reflecting the difficulty of coercing nearby observations into the paradigm. However, the site precision π_n remains large

from the initial inclusion, and the cavity variance $\sigma^2_{\backslash n} = (\sigma_n^{-2} - \pi_n)^{-1}$ may as a result be negative. In this case, we have little option but to skip the site and hope a refinement is possible after subsequent iterations have stabilised the posterior. Alternatively, we can restart the inference and use a higher damping factor in an effort to prevent the premature shrinking of the posterior at the observation. When the outliers appear in clusters, or when they possess some inherent structure, these problems are exacerbated since local smoothness properties are less violated upon their inclusion as "real", and there is even less evidential support for subsequently relegating them to outlier status.

We emphasise that for the mixture model, such clusters are generally neither necessary nor sufficient to cause convergence to the wrong mode: the problem is created primarily by their inclusion *at the early stages* of the first EP loop—when the posterior approximation is still akin to the rather agnostic prior—and the consequent collapse of the posterior onto the erroneous data. The left-hand column of fig. 3.12 illustrates its behaviour when a cluster of outliers is introduced into the data; we find EP readily adopts them as truth and essentially forgets the context of the observations provided by the prior. The other columns illustrate the more reserved response of the TGP, and will be discussed below.

There are two related issues to recognise about the i.i.d. mixture. First, EP does not retain sufficient uncertainty about the marginal distribution; early decisions made too firmly cannot easily be undone. Second, there is no communication between data about their reliability; particularly if we suspect that outliers may occur in clusters, then detecting the presence of one should make us wary of its neighbours.

Let us consider these points in turn. We have already described how i.i.d. mixture noise is a special case of the TGP: it corresponds to using everywhere a constant value for $\mathbf{u}$. However, if we allow a more general process and regard the $\mathbf{u}$ as "level 1" parameters (like $\mathbf{f}$), to be *marginalized*, not optimized, then use of the standard mixture seems slightly curious. It is somewhat reminiscent of the historical debate over how to deal with the lower parameters of a Bayesian model: MacKay (1999) warns of the dangers of optimizing (particularly after marginalizing out the hyperparameters, as is done for example in Buntine and Weigend (1991)), since this often locates a filamentary peak in the likelihood, highly unrepresentative of the bulk of posterior mass. Of course, in the Gaussian mixture, the solution is heavily regularized since we set only a single e, not the vector $\mathbf{u}$. However, referring again to fig. 3.12, the central column illustrates the *joint* distribution over u and f when multiple outliers are introduced, and in these

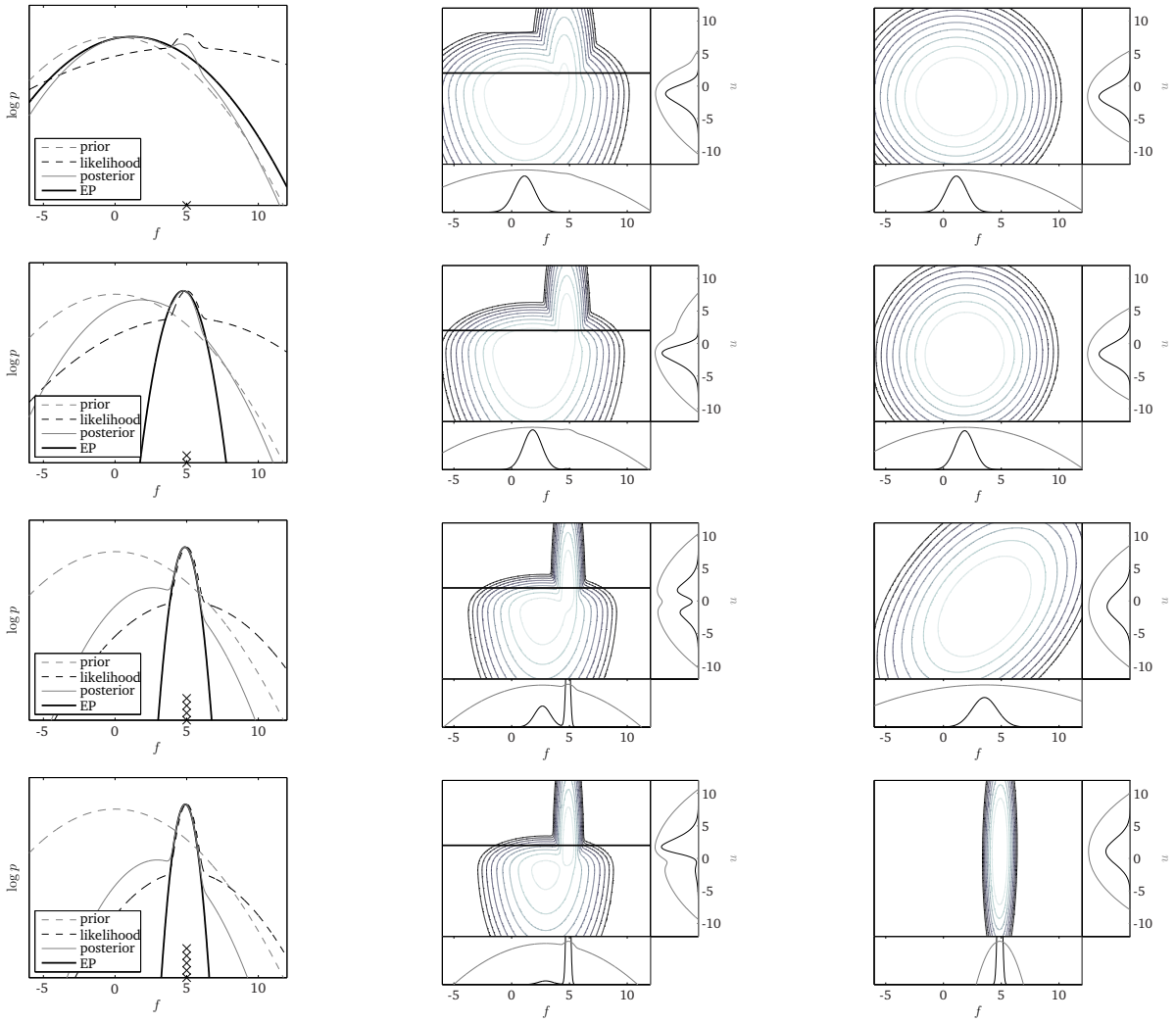


Figure 3.12: Using the twinned Gaussian process provides a natural resilience against clustered noisy data. The left-hand column illustrates the behaviour of a fixed heavy-tailed likelihood for one, two, four and five repeated observations at $f = 5$. (Outliers in real data are not necessarily so tightly packed, but the symmetry of this approximation allows us to treat them as a single unit: by "posterior", for example, we mean the a posteriori belief in *all* the observations' (identical) latent f.) The context is provided by the prior, which gives 95% confidence to data around $f = 0 \pm 2$. The top-left box illustrates how the influence of isolated outliers is mitigated by the standard mixture. However, a repeated observation (box two on the left) causes the EP solution to collapse onto the spike at the data (the log scale is deceptive: the second peak contributes only about 8% of the posterior mass). The twinned GP better preserves the marginal distribution of f by maintaining a joint distribution over both f and u: in the second and third columns respectively are contours of the true log joint (we use a broad zero-mean prior on u) and that inferred by EP, together with the marginal posterior over f. Only with a fifth observation—final box—is the context of f essentially overruled by the TGP approximation. The thick bar in the central column marks the cross-section corresponding to the unnormalized posterior from column one.

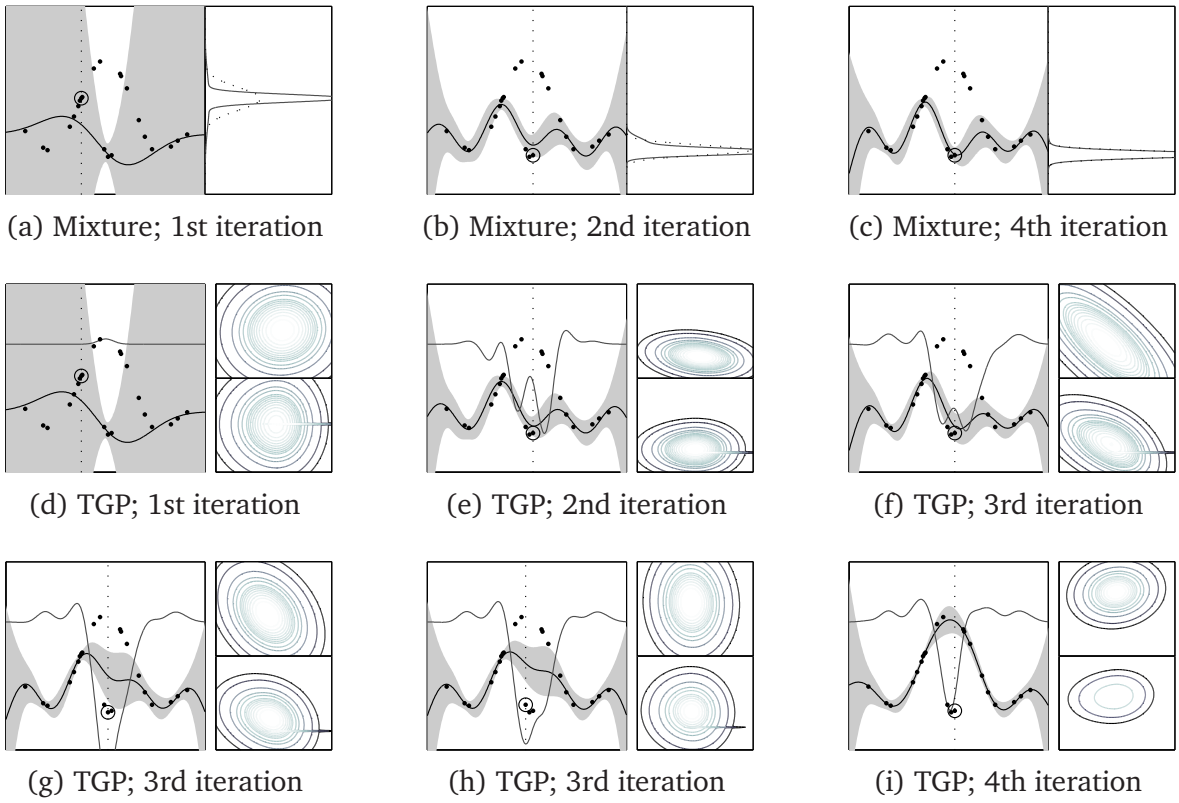
figures we find just such a spike of probability. Observe too that the global distribution does not necessarily have two well-separated modes, even if a slice through it at fixed u does. Notice further that for fixed $u = 2$ (corresponding to an assumption that about 2.5% of observations are outliers), the cross-sections (first column) are generally a poor representation of the full distribution. Finally, in the third column, witness the effect of using EP with a *bivariate* Gaussian to match moments. A welcome consequence of deferring marginalization of u until prediction time is that the algorithm remains agnostic about the distribution of f until more data are considered.

By placing a full GP on $\mathbf{u}$ we have the further benefit that information about outliers can propagate through the data in a manner impossible with i.i.d. methods. Thus, even if a firm assignment happens to be made incorrectly (see final plot in fig. 3.12)—a decision which can irrevocably damage convergence under the i.i.d. assumption—the TGP provides a natural mechanism for its reversal provided the distribution on $\mathbf{u}$ has not also collapsed.[8] In the neighbourhood of outliers mistaken for "real" data, any genuine observations are likely themselves to be mislabelled as outliers since they violate the current hypothesis. However as a result, the $\mathbf{u}$-process is encouraged to lend weight to the outlier component of the mixture, broadening the noise model in their vicinity. This in turn can provide a bridge to the more probable mode of the posterior, in which all these data receive their correct interpretation. This process of revision is illustrated in fig. 3.13, and contrasted with the results of using the conventional mixture model, in which no such recovery may be possible.

3.5.2 Conclusions

We argue that a two-component Gaussian mixture is a sensible model for many real data, with a natural interpretation and the heavy tails required for robustness, whose weaknesses are exposed primarily when the noise distribution is not homoscedastic. The TGP extends the applicability of the standard mixture to such difficult cases while retaining tractability with respect to EP—significantly faster than the heavy duty Monte Carlo methods required for the more complex models. It can be viewed both as a generalization of the mixture, and as a specialization of flexible GP mixtures developed by Tresp (2001) and Rasmussen and Ghahramani (2002).

[8] This is unlikely to occur unless the prior is made too restrictive, since there is generally insufficient evidence from the observations themselves to describe the distribution on $\mathbf{u}$ with great certainty.



(a) Mixture; 1st iteration (b) Mixture; 2nd iteration (c) Mixture; 4th iteration

(d) TGP; 1st iteration (e) TGP; 2nd iteration (f) TGP; 3rd iteration

(g) TGP; 3rd iteration (h) TGP; 3rd iteration (i) TGP; 4th iteration

Figure 3.13: In the top row, the GP with i.i.d. mixture noise converges to a comparatively poor solution, because adversarially we have forced it to consider initially the three outliers in the data. In the right-hand plot of each of these panels is, in solid grey, the true posterior on f, and dotted, the Gaussian which matches its moments. The lower two rows show the behaviour of the TGP model with an identical ordering of site refinement. Initially, it too finds the inferior approximation (e), but now there is the implication of an implausible kink in $\mathbf{u}$ at the outlying data. In the third iteration of EP, the kink is ironed out, and by the start of the fourth, the posterior has converged to the desired solution. The right-hand plots of these panels show the bivariate approximation (above) and the true posterior (below), with u on the horizontal axis, and f vertically. Observe the slender ridge of high probability for large u in the lower plot, which considered globally has insufficient mass to capture the Gaussian approximation.

With respect to the conventional model, we have illustrated how convergence in the TGP may be more reliable under EP, and experimentally we have exhibited its superior ability to model uncertainty in the latent signal: even in domains for which the mixture can perform well (such as homoscedastic corruption with infrequent, isolated outliers), the predictive performance of the TGP is significantly stronger. For more general heteroscedasticity, the mixture is obviously straitened by its i.i.d. assumption; furthermore, prior knowledge about the noise distribution can be incorporated readily into the TGP. On the other hand, for data in which the sample variance changes over

several orders of magnitude, we have found the TGP to be rather inadequate. With respect to the more ambitious models, we believe ours is a valuable addition to the library of GP mixtures: on a qualitative plot of error rate against prediction time, we envisage it appearing somewhere near the lower-left corner of the axes. Although the mixture of Rasmussen and Ghahramani drives the error lower, it is at the expense of considerably increased time complexity.

CHAPTER 4

Extending the twinned Gaussian process

THE TWINNED GAUSSIAN process (TGP) presented in chapter 3 was devised as a model for efficient robust regression. In this chapter we aim to illustrate how the TGP constitutes a flexible class of Gaussian process models that share the property of tractable EP inference. In section 4.1, we consider a robust approach to binary classification; section 4.2 presents an alternative noise model for robust regression, while the remaining three sections describe approaches to mixture modelling motivated by the TGP framework.

4.1 Robust classification

In the regression domain of chapter 3, we used the term "outlier" to refer to those observations which grossly violate the structure of data in their vicinity. The light tails of a Gaussian likelihood were unsuitable, and an alternative noise model was proposed. In the context of binary classification the equivalent notion is less well defined, where "outliers" can only manifest themselves as mislabelled data. If we encounter clusters of apparently mislabelled data, our indirect knowledge of the latent process via binary assignments means we are likely to be less confident of their outlier status than in the regression case because the continuous latent process is less directly

related to the discrete observations. In fact, the classical probit noise model is already somewhat robust to labelling errors since any finite value on the latent $\mathbf{f}$ process yields a non-zero (although potentially very small) probability for either class assignment. In this case, erroneous data in sufficiently sparse quantities have the effect of moderating the latent signal such that predictions are made with less certainty.

For these reasons, we do not envisage an extension to classification of the TGP to be employed widely in practice, but present the derivation here for two reasons. First, it is possible that such a model would find application in a noisy classification task; the behaviour of the $\mathbf{u}$ process may itself be of interest, for example, if we believe a priori that corruptions are arising at some frequency we seek to establish. Second, the extension is of theoretical interest in its own right, since it serves as a convenient introduction to an approximation also employed in section 4.5, where a secondary EP loop calculates intractable moments of the tilted distribution.

4.1.1 The model

Following the ideas of the TGP, we introduce an auxiliary GP on $\mathbf{u}$ that will form a soft partition of the domain into two generative processes. A flexible likelihood distribution is used,

$$p(y_n = +1|u_n, f_n, \boldsymbol{\pi}) = \sigma(u_n)\sigma(\pi_R f_n) + \sigma(-u_n)\sigma(\pi_O f_n),$$

with additional parameters $\boldsymbol{\pi}$ to control the slope of the sigmoid function on f_n. In the limit $\pi_R \to \infty$, the "real" process behaves like a step function; when $\pi_O \to 0$, the "outlier" process assigns labels equiprobably and independently of f_n. The introduction of $\boldsymbol{\pi}$ is crucial: without different factors in evaluating the second pair of sigmoids, certain derivatives in the EP loop cancel and prevent the inference of any useful process on $\mathbf{u}$ whatsoever.

In conjunction with a bivariate Gaussian prior and after renormalizing, we obtain the joint distribution, an example of which appears in fig. 4.1. It is qualitatively of a similar shape to the marginal posterior in the original TGP (fig. 3.6c); again, we observe that in general there will be a posteriori correlations between u_n and f_n.

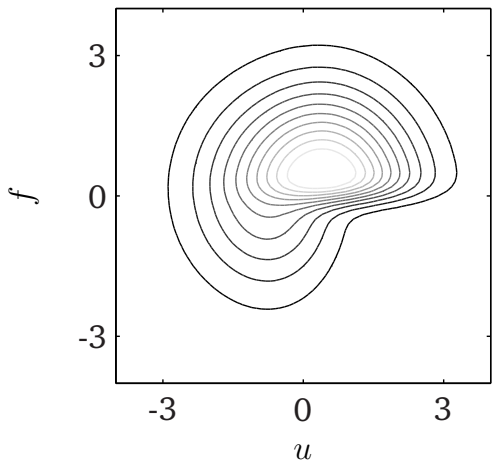


Figure 4.1: The marginal posterior at a positively-labelled example. Observe that for large positive u, the latent f is almost certainly greater than zero, whereas if u is negative the distribution on f is rather broad.

4.1.2 Inference

Implementation of an expectation propagation (EP) inference procedure requires a closed form for each

$$Z_n^\lambda = \iint \sigma(u_n)\sigma(\pi_\lambda f_n)\mathcal{N}\left(\begin{bmatrix} u_n \\ f_n \end{bmatrix}; \boldsymbol{\mu}^{\backslash n}, \boldsymbol{\Sigma}^{\backslash n}\right) \mathrm{d}u_n \mathrm{d}f_n, \tag{4.1}$$

with $\lambda \in \{R, O\}$, to derive $\boldsymbol{\mu}_\lambda = \mathbb{E}\begin{bmatrix} u_n \\ f_n \end{bmatrix}$ and $\boldsymbol{\Sigma}_\lambda = \mathbb{V}\begin{bmatrix} u_n \\ f_n \end{bmatrix}$.

Unfortunately, no analytic solution exists. We could rewrite (4.1) as an integral over a bivariate Gaussian by expanding the σ functions, after which stochastic methods can be employed to calculate approximations to Z_n, $\boldsymbol{\mu}_\lambda$ and $\boldsymbol{\Sigma}_\lambda$; see for example Genz (2004). However, a more efficient and purely deterministic solution presents itself if we identify (4.1) as the marginal likelihood of a standard GP classification model with a data set of size two: a secondary EP loop (or library call to a GP classifier) then yields approximations to $Z_n, \boldsymbol{\mu}_\lambda$ and $\boldsymbol{\Sigma}_\lambda$. They can be plugged into expressions from the primary EP loop (see section 1.3); by the chain rule:

$$\begin{aligned} \boldsymbol{\alpha}_n = \frac{\partial \log Z_n}{\partial \boldsymbol{\mu}^{\backslash n}} &= \frac{1}{Z_n}\left(\frac{\partial Z_n^R}{\partial \boldsymbol{\mu}^{\backslash n}} + \frac{\partial Z_n^O}{\partial \boldsymbol{\mu}^{\backslash n}}\right) \\ &= (\boldsymbol{\Sigma}^{\backslash n})^{-1}\left(\frac{Z_n^R \boldsymbol{\mu}_R + Z_n^O \boldsymbol{\mu}_O}{Z_n^R + Z_n^O} - \boldsymbol{\mu}^{\backslash n}\right), \end{aligned} \tag{4.2}$$

and

$$\boldsymbol{\nu}_n = -\frac{\partial^2 \log Z_n}{\partial \boldsymbol{\mu}^{\backslash n} \partial \boldsymbol{\mu}^{\backslash n T}} = \left(\frac{\partial \log Z_n}{\partial \boldsymbol{\mu}^{\backslash n}}\right)^2 - \frac{1}{Z_n}\left(\frac{\partial^2 Z_n^R}{\partial \boldsymbol{\mu}^{\backslash n} \partial \boldsymbol{\mu}^{\backslash n T}} + \frac{\partial^2 Z_n^O}{\partial \boldsymbol{\mu}^{\backslash n} \partial \boldsymbol{\mu}^{\backslash n T}}\right) \tag{4.3}$$
$$= \boldsymbol{\alpha}\boldsymbol{\alpha}^T - \frac{1}{Z_n^R + Z_n^O}(\boldsymbol{\Sigma}^{\backslash n})^{-1}\left(Z_n^R \mathbf{B}_R + Z_n^O \mathbf{B}_O\right)(\boldsymbol{\Sigma}^{\backslash n})^{-1},$$

where $\mathbf{B}_\lambda = \boldsymbol{\Sigma}_\lambda + \boldsymbol{\mu}_\lambda(\boldsymbol{\mu}_\lambda - \boldsymbol{\mu}^{\backslash n})^T + \boldsymbol{\mu}^{\backslash n}(\boldsymbol{\mu}^{\backslash n} - \boldsymbol{\mu}_\lambda)^T - \boldsymbol{\Sigma}^{\backslash n}$.

Let us pause for a moment: recall that by making a Gaussian approximation, EP is actually ignoring the asymmetry in the marginal posterior that arises from the product of a Gaussian and two sigmoids. However the original problem (finding a Gaussian approximation to the full posterior over $\mathbf{u}$ and $\mathbf{f}$) requires only the zeroth, first and second moments at each site; hence, provided their estimates from the inner EP loop are sufficiently accurate—which empirically, they certainly are—the intractability of (4.1) is not a critical concern. Furthermore, although we are assured only of *termwise* moment matching, in this case there are always precisely two sites involved, and we may expect that the global approximation is faithful.

We note that the restricted form of (4.1) means we can use a relatively pared down GP classifier for $\boldsymbol{\alpha}_n$ an $\boldsymbol{\nu}_n$; convergence tests, for example, are unnecessary. This inner EP loop forms the bottleneck in the main inference procedure, and an implementation would benefit from a precompiled subroutine.

Model selection

Derivatives of the marginal likelihood are required to optimize hyperparameters of the model. Those relating to parameters of the prior (of the kernel function, for example) can be obtained as before. However, since the moment contributions of the individual sites no longer exist in closed form, derivatives corresponding to hyperparameters of the likelihood (such as the $\boldsymbol{\pi}$) require a different treatment. Let α be such a variable and let L denote the log marginal likelihood. At convergence (see section 1.3), we have

$$\frac{\partial L}{\partial \alpha} = \sum_{n=1}^{N} \frac{\partial \log Z_n}{\partial \alpha}, \tag{4.4}$$

where $Z_n = Z_n^R + Z_n^O$ is the zeroth moment of the tilted distribution at site n, and is the sum of two terms like (4.1). Commonly, the right-hand side of (4.4) can be calculated directly, but in this setting there is only an EP estimate of Z_n and no immediate means

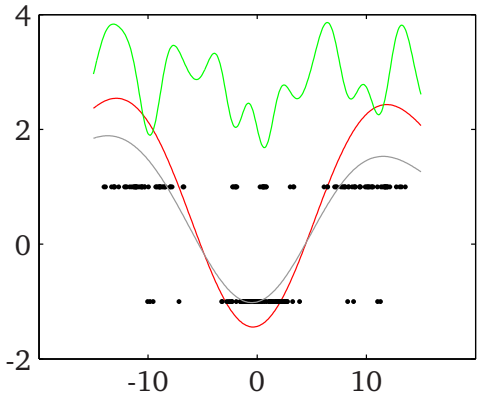


Figure 4.2: The data appear as black dots, divided at ± 1 into two classes. In feint grey is the solution obtained using standard GP classification (a probit model, trained by EP). The TGP classifier appears in red, with the associated $\mathbf{u}$ process oscillating at high frequency in green. Observe that the red curve is more confident in its class assignments, and that the troughs of the green curve largely coincide with anomalies in the data.

to calculate the derivative. Fortunately, we can again apply the chain rule: if α appears in Z_n^λ only at subsite j (i.e. one of the sigmoids in (4.1)),

$$\frac{\partial \log Z_n}{\partial \alpha} = \frac{1}{Z_n}\frac{\partial Z_{n,j}^\lambda}{\partial \alpha} = \frac{1}{Z_n}\frac{\partial Z_{n,j}^\lambda}{\partial \log Z_{n,j}^\lambda}\frac{\partial \log Z_{n,j}^\lambda}{\partial \alpha} = \frac{Z_{n,j}^\lambda}{Z_n}\frac{\partial \log Z_{n,j}^\lambda}{\partial \alpha},$$

and the derivative in the final expression is analytically tractable.

4.1.3 Experiments

As proof of concept and for easy visualization, we consider a unidimensional toy set in which the training labels have been flipped according to the oscillations of a high-frequency latent sinusoid (fig. 4.2). After training, we find the TGP extended to classification is able to give more confident predictions than the standard GP classifier since labels which would otherwise heavily penalize the marginal likelihood can be relegated to the outlier component. The GP classifier in contrast must moderate the amplitude of its latent process globally in order to compensate for the outliers.

In addition, the $\mathbf{u}$ process (here with a squared exponential kernel) provides a description of the behaviour of the corruptions. We remark that in certain fields, it is precisely these rare and anomalous labels that are of interest to the investigator, in which case the TGP gives significantly more useful feedback than the standard model.

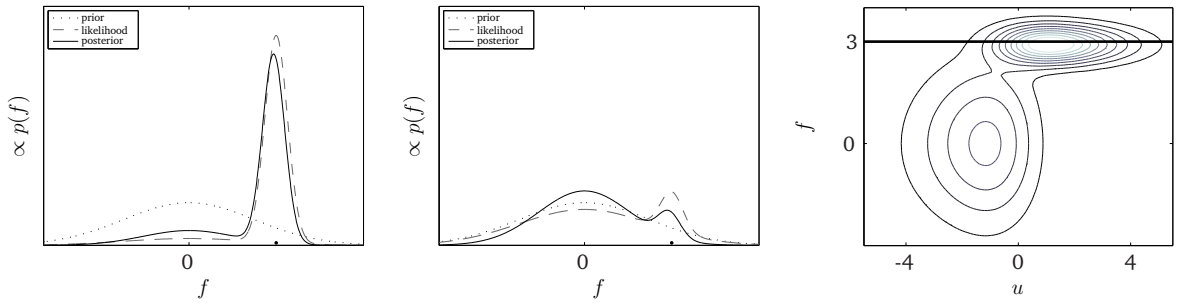


Figure 4.3: In panels (a) and (b), the likelihood of the "ignorance model" for fixed u. These are cross-sections at $u = \pm 1$ of panel (c), the posterior after observing $y = 3$; in this example, we have set $f_0 = 0$. For a comparison with the conventional TGP mixture, refer to fig. 3.6.

4.2 A model for ignorance

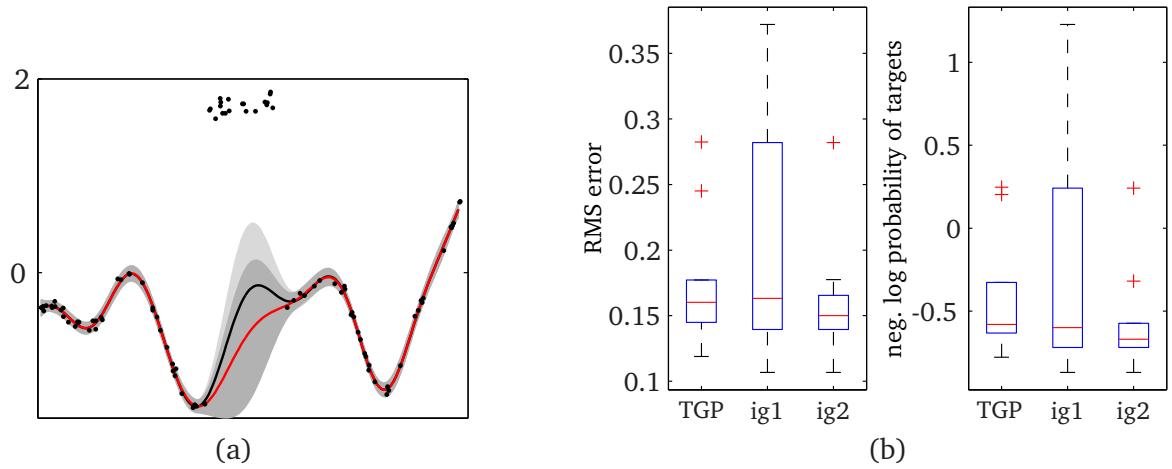
The TGP noise model (3.5) used in chapter 3 demands that outliers are evenly distributed around the latent f, albeit with the possibility of large variance. If, however, outlying observations in the data do not appear as large-variance corruptions of f, but rather as if drawn from some secondary distribution, we may prefer to use an outlier model that more closely accords with our prior beliefs. One option would be to fix the mean at some value f_0, learned as an additional parameter, such that

$$y_n = \begin{cases} \mathcal{N}(y_n\,;\,f_0\,,\,\sigma_O^2) & \text{with probability } e_n = 1-\sigma(u_n), \\ \mathcal{N}(y_n\,;\,f_n\,,\,\sigma_R^2) & \text{with probability } 1-e_n = \sigma(u_n). \end{cases}$$

Because dependencies between y_n and f_n are eradicated entirely as e_n approaches 1, the behaviour of this noise model is to ignore data deemed to have originated from the outlier component. The marginal posterior is illustrated in fig. 4.3.

Inference for this model proceeds identically to the TGP, except that obtaining moments from the outlier component is more straightforward due to its f-independence:

$$\begin{aligned} Z^O &= \iint \sigma(-u)\mathcal{N}\big(y\,;\,0\,,\,\sigma_R^2\big)\,\mathcal{N}\left(\begin{bmatrix} u \\ f \end{bmatrix};\begin{bmatrix} \mu_u \\ \mu_f \end{bmatrix},\begin{bmatrix} \sigma_{uu}^2 & \sigma_{uf}^2 \\ \sigma_{fu}^2 & \sigma_{ff}^2 \end{bmatrix}\right)\mathrm{d}u\mathrm{d}f \\ &= \mathcal{N}\big(y\,;\,0\,,\,\sigma_O^2\big)\int_u \sigma(-u)\mathcal{N}\big(u\,;\,\mu_u\,,\,\sigma_{uu}^2\big)\,\mathrm{d}u \\ &= \mathcal{N}\big(y\,;\,0\,,\,\sigma_O^2\big)\,\sigma\left(-\frac{\mu_u}{\sqrt{1+\sigma_{uu}^2}}\right). \end{aligned}$$



(a) (b)

Figure 4.4: In panel (a), the ignorance noise model is applied to toy data; its inferred mean appears in red, while that of the original TGP is in black. Panel (b) shows the model applied to ten folds of the set Friedman (2). On two folds, the results were very poor; these have been included in "ig1", but omitted from "ig2".

The only non-zero derivatives are

$$\frac{\partial Z^O}{\partial \mu_u} = -\frac{\mathcal{N}(y\,;\,0\,,\,\sigma_O^2)\,\mathcal{N}\left(\frac{\mu_u}{\sqrt{1+\sigma_{uu}^2}}\right)}{\sqrt{1+\sigma_{uu}^2}}; \qquad \frac{\partial^2 Z^O}{\partial \mu_u^2} = \frac{\mu_u \mathcal{N}(y\,;\,0\,,\,\sigma_O^2)\,\mathcal{N}\left(\frac{\mu_u}{\sqrt{1+\sigma_{uu}^2}}\right)}{1+\sigma_{uu}^2}.$$

4.2.1 Experiments

To illustrate how the "ignorance" model can differ from the original TGP, its behaviour on unidimensional toy data is shown in fig. 4.4a. Since corruptions in the data are now explained as Gaussian noise spread around a fixed value (in this case arbitrarily fixed at zero, although f_0 can easily be learned as an extra hyperparameter), once data have been classified as outliers they exert no pull on the latent mean, which can more smoothly interpolate the remaining data. The parameters have been tied between models in this example, and chosen to accentuate their differences; after training, the contrast is less pronounced since with a sufficiently broad "outlier" distribution, the pull on the TGP interpolant can be made very small. In the general case, more varied data in other regions of the input may prevent this adaptation.

We also evaluate performance on the second Friedman set introduced in section 3.3. Corruptions there were clustered distantly from the underlying signal; the expectation is that explicitly ignoring them will improve accuracy. In practice (see fig. 4.4b), the

ignorance model performed very poorly for two of the ten folds, consistently converging on a markedly inferior solution: our belief is that it confidently ignores isolated data which in reality are part of the latent signal, in a manner that the original TGP did not. Interestingly, the two folds on which the TGP performed worst were amongst those on which the ignorance model performed best, so it is difficult to make a universal statement about suitability. However, with such variability in results, it would seem that the "ignorance" paradigm is a rather dangerous one!

4.3 Mixtures of Gaussian processes

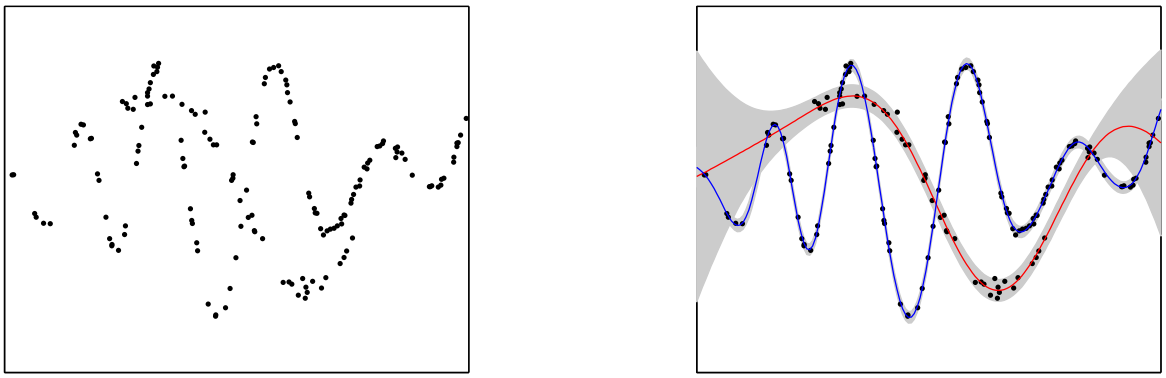
We consider the following scenario. Measurements of the heights and weights of a series of individuals are made, and a broadly positive correlation is observed. However, it is also noticed that the data may best be explained by two models; although they have not been annotated, we presume these correspond to men and women. In this example (and related settings), we might attempt to fit a single regression curve, i.e. a solution for the generic human. Alternatively, we could ignore portions of the data as "outliers", attempting to focus on either male or female—in this case, there is no reason to expect the clustering behaviour the TGP was designed to address, and a standard heavy-tailed model may work (although it will likely be difficult to establish which points to "ignore"). The ideal solution is probably to fit two regression functions simultaneously.

We describe next how the TGP can be extended very easily to mimic a two-component mixture, but avoiding any requirement for stochastic sampling (Tresp, 2001): inference is by EP. With this perspective, the notion of "outlier" is discarded: there is a symmetry between the two generative models, with one GP on latent $\mathbf{f}$, and another on $\mathbf{g}$; noise is now i.i.d., although its variance may differ between the two processes. We still employ a set of latent variables $\mathbf{u}$, but they are now a priori independent since our assumptions have changed: we expect to observe two overlapping functions, rather than a dominant function with (clustered) outliers. In summary,

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}\,;\,\mathbf{0}\,,\,\mathbf{K_{ff}})\,; \qquad p(\mathbf{g}|\mathbf{X}) = \mathcal{N}(\mathbf{g}\,;\,\mathbf{0}\,,\,\mathbf{K_{gg}})\,; \qquad p(\mathbf{u}) = \mathcal{N}(\mathbf{u}\,;\,\mathbf{m_u}\,,\,\mathbf{I})\,.$$

The likelihood is

$$p(y_n|f_n, g_n, u_n) = \sigma(u_n)\mathcal{N}\big(y_n\,;\,f_n\,,\,\sigma_f^2\big) + \sigma(-u_n)\mathcal{N}\big(y_n\,;\,g_n\,,\,\sigma_g^2\big)\,. \tag{4.5}$$



(a) Training data (b) Inferred generative processes

Figure 4.5: Extending the TGP to a mixture of GPs.

There are three GPs to deal with, hence space and time requirements grow to $\mathcal{O}\left((3N)^2\right)$ and $\mathcal{O}\left((3N)^3\right)$ respectively, but no new mathematics is introduced beyond the original TGP. The symmetry in the likelihood allows both components to be treated essentially the same (being careful to account for $-u_n$ in the second); moment calculations and estimates of the marginal likelihood and its derivatives all follow without alteration or complication. After we have run the EP inference procedure, the only processes likely to be of interest are on $\mathbf{f}$ and $\mathbf{g}$: since a priori we assumed no correlation in $\mathbf{u}$, there can be none in the posterior.

4.3.1 Experiments

By way of example, we drew unidimensional data from two GP priors with different covariances and combined them into a single set (see fig. 4.5a). Due to the overlap, there is some difficulty in distinguishing which points belong to which generative component, but using our GP mixture model, the distinction is made clear (see fig. 4.5b), and orders of magnitude more quickly than by sampling from the posterior with MCMC.

To investigate the reliability of convergence on higher dimensional problems, we tested our model on toy data generated by combining the output of two independent GPs. Training ordinates were sampled from the ten-dimensional unit cube, with between 50 and 100 points from each source, with the source processes using the anisotropic squared exponential covariance initialized with random parameters (each log length-scale was drawn from a unit-variance Gaussian with mean -2; the log amplitude was

drawn from a unit-variance Gaussian with mean 1). To the latent values was added i.i.d. noise of variance 0.01. When the hyperparameters were intialized close to those of the generative model, the iterations converged without fail for ten instantiations of the data, achieving error rates identical to those obtained by training two separate GPs (in conjunction with an "oracle" labelling each input with its source component). This confirms that EP is theoretically able to fit the latent processes optimally. However, the multimodality of the posterior caused problems when we attempted to optimize randomly-initialized hyperparameters. Fitting an arbitrary mixture without prior knowledge is certainly a difficult problem, and we conclude that for our method, training should begin at several trial initializations, from which can be chosen that model with greatest evidential support.

4.3.2 Variational methods

An alternative paradigm for deterministic inference is provided by the variational approach, introduced briefly in section 1.4. It enjoys certain advantages; for example, in contrast to EP, convergence is guaranteed, in this case to some local optimum of the Kullback-Leibler divergence between approximating and true distributions. We also obtain a strict lower bound on the true evidence. Furthermore, in the case of the mixture model presented above, the extension to an arbitrary number of components *in fixed proportions* is relatively straightforward, by placing a Dirichlet prior on their weights. However, in practice the inference is also sensitive to intialization of the latent assignments: although we are assured of convergence it is by no means necessarily to a global optimum (with respect to the posterior belief in each datum's generative component).

We must make clear that variational methods cannot be applied as easily to the TGP as they can to mixtures of GPs with unknown but *fixed* weights. To make this clear, consider attempting a variational implementation: first, we introduce latent allocation variables $\mathbf{c}$, so the joint distribution is

$$p(\mathbf{y}, \mathbf{c}, \mathbf{f}, \mathbf{u}) = \prod_{n=1}^{N} \left[\sigma(u_n)\mathcal{N}\left(y_n\,;\, f_n\,,\, \sigma_R^2\right)\right]^{\frac{1}{2}(1+c_n)} \left[\sigma(-u_n)\mathcal{N}\left(y_n\,;\, f_n\,,\, \sigma_O^2\right)\right]^{\frac{1}{2}(1-c_n)} p(\mathbf{f})p(\mathbf{u}),$$

where $c_n \in \{\pm 1\}$ indexes "real" and "outlier", and the likelihoods $p(y_n|f_n)$ and priors $p(\mathbf{u})$ and $p(\mathbf{f})$ are all Gaussian. We make a factorizing approximation (which, we note

in passing and with reference to fig. 3.6c, seems rather counterintuitive),

$$Q(\mathbf{c}, \mathbf{f}, \mathbf{u}) = q_c(\mathbf{c}) q_f(\mathbf{f}) q_u(\mathbf{u}),$$

and alternate VBE steps to update q_c, and VBM steps to update q_f and q_u:

$$\begin{aligned} \textbf{VBE} \quad & q_c(c_n) \propto \exp\left(\langle \ln p(y_n, c_n | f_n, u_n) \rangle_{q_f(f_n), q_u(u_n)}\right); \\ \textbf{VBM} \quad & q_f(\mathbf{f}) \propto p(\mathbf{f}) \exp\left(\langle \ln p(\mathbf{y}|\mathbf{c}, \mathbf{f}, \mathbf{u}) \rangle_{q_c(\mathbf{c}), q_u(\mathbf{u})}\right); \\ & q_u(\mathbf{u}) \propto p(\mathbf{u}) \exp\left(\langle \ln p(\mathbf{y}|\mathbf{c}, \mathbf{f}, \mathbf{u}) \rangle_{q_c(\mathbf{c}), q_f(\mathbf{f})}\right). \end{aligned}$$

Since c_n is binomially distributed and appears in linear form only, expectations with respect to the $\mathbf{c}$ are tractable. Similarly tractable are expectations with respect to $\mathbf{f}$. Unfortunately, we find intractable expectations $\int \mathcal{N}(u_n) \ln \sigma(u_n) \mathrm{d}u_n$ arise in both steps of the algorithm, and these require further approximations. One option is to use the logit rather than the probit and apply an exponential bound to the link function (Jaakkola and Jordan, 1996); the approach was employed successfully for binary classification in GPs by Gibbs and MacKay (2000). Indeed, if we write the joint as

$$p(\mathbf{y}, \mathbf{c}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f}, \mathbf{c}) p(\mathbf{c}|\mathbf{u}) p(\mathbf{f}) p(\mathbf{u}) = \mathcal{N}(\mathbf{y}\,;\,\mathbf{f}\,,\,\mathbf{S}) \left(\prod_{n=1}^{N} p(c_n|u_n) \right) p(\mathbf{f}) p(\mathbf{u}),$$

where the diagonal matrix $\mathbf{S}$ was defined in (3.8), the difficult term $p(\mathbf{z}|\mathbf{u})$ is a product of sigmoids, and in conjunction with the prior on $\mathbf{u}$ is isomorphic to the GP classification problem (we also observed this relationship in the Monte Carlo evaluation of section 3.4). Alternatively, we might instead only consider working with moments of the distribution on $\mathbf{u}$. Since we know EP converges reliably for binary classification and with excellent results, they can be obtained approximately in that manner for a given set of latent assignments $\mathbf{c}$. We defer the development of such ideas, and the more general investigation of a variational approximation to the TGP posterior, to future work.

4.4 Mixtures of two experts

Allowing for a more general prior on $\mathbf{u}$ (together with priors on $\mathbf{f}$ and $\mathbf{g}$) recovers an instance of the classic *mixture of experts* architecture (Jacobs et al., 1991), which attempts to divide a learning problem into a series of sub-problems and conquer each

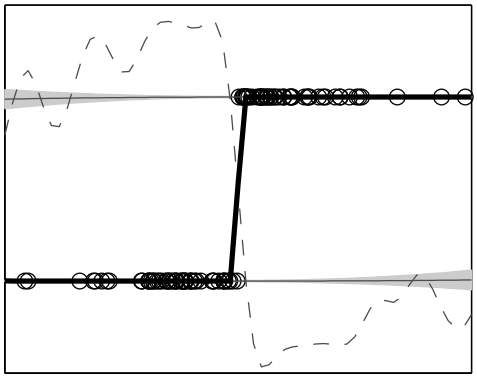


Figure 4.6: Employing a mixture of experts architecture to approximate the step function. The result appears in thick black; the training data are hollow circles, and the two source processes appear in feint grey; the dashed line is the gating process.

individually with a specialized model. The gating process is used to select the appropriate regressor based on the location of the test input alone: here, the $\mathbf{u}$ process occupies the role. We use the likelihood (4.5), and the priors are

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}\,;\,\mathbf{0}\,,\,\mathbf{K_{ff}})\,; \quad p(\mathbf{g}|\mathbf{X}) = \mathcal{N}(\mathbf{g}\,;\,\mathbf{0}\,,\,\mathbf{K_{gg}})\,; \quad p(\mathbf{u}) = \mathcal{N}(\mathbf{u}\,;\,\mathbf{m_u}\,,\,\mathbf{K_{uu}})\,,$$

Tresp (2001) gives the example of using a GP mixture model to learn a step function from squared exponential kernels. The solution in that paper uses three GPs (plus the additional gating processes) to fit regions around the transition, with a very high frequency function modelling the critical region, and lower frequencies employed further from it. The learning procedure employs Monte Carlo simulation. For comparison, we applied our model to the same problem, using two GPs and the gating process; our results are illustrated in fig. 4.6. The processes have arranged themselves to fit either half of the step, treating them nearly as constants by increasing the lengthscale. A short-wavelength GP on $\mathbf{u}$ then makes the sharp transition between the experts. (In fact, this problem can be solved easily using a more suitable covariance function; see Rasmussen and Williams (2006, sec. 5.4).)

We note that there is a computational issue in this essentially zero-noise regime, since underflow can be a problem while EP is trying to settle on a solution. It is necessary for convergence to make an initial run with a rather wide noise distribution ($\sigma_f^2 = \sigma_g^2 = 0.01$), retaining the resulting approximation into a subsequent iteration of EP for which the variances can be reduced to 10^{-5}.

4.5 Enriching the outlier process

We have argued that the two-component mixture is a sensible model for real world data in which non-Gaussian corruptions may arise. The principal component models the underlying function, while the secondary process is employed to explain any implausible large-variance deviations. Occasionally, this model may be too restrictive. It was explained in section 3.3.1 how the TGP struggles accurately to model heteroscedastic data with a very large range in variance. A second problem may arise in the simple assumption of a unimodal error distribution. Consider for example when outliers arise not only as large-variance corruptions of the latent signal (due perhaps to the heavy tails of the corrupting process), but also in the form of section 4.2, highly correlated outputs that are independent of the modelled signal (this systematic corruption may result from a faulty measurement device). Such variability in the behaviour of the nuisance noise is a potential problem for any unimodal likelihood, since it must explain simultaneously clusters away from the signal and widely spread pure noise scattered around it. This is a motivating example, but the extension we propose in this section is applicable wherever just two components prove insufficient. In fact, the idea appears in its most general form as an arbitrary mixture of GP experts in which all inference is conducted by EP.

4.5.1 The model

Concentrating initially on the question of a complex outlier distribution, there are two avenues we can explore. The first introduces a compound likelihood for the second component

$$p_O(y_n|f_n) = \sum_{c=1}^{C} \alpha_c \mathcal{N}\left(y_n\,;\,\mu_c(f_n)\,,\,\sigma_c^2\right); \qquad \sum_{c=1}^{C} \alpha_c = 1; \tag{4.6}$$

the arrangement is illustrated in fig. 4.7a. In this case, the inference of section 3.2 applies with only small changes: each component of p_O contributes moments independently, and EP proceeds with the slight extra cost of these additional calculations at each iteration.

There are certain drawbacks. First, the generative model sacrifices correlations in the data for a simple representation, by merging two or more distributions into a single Gaussian mixture: the $\mathbf{u}$-process encourages the formation of clusters of outliers

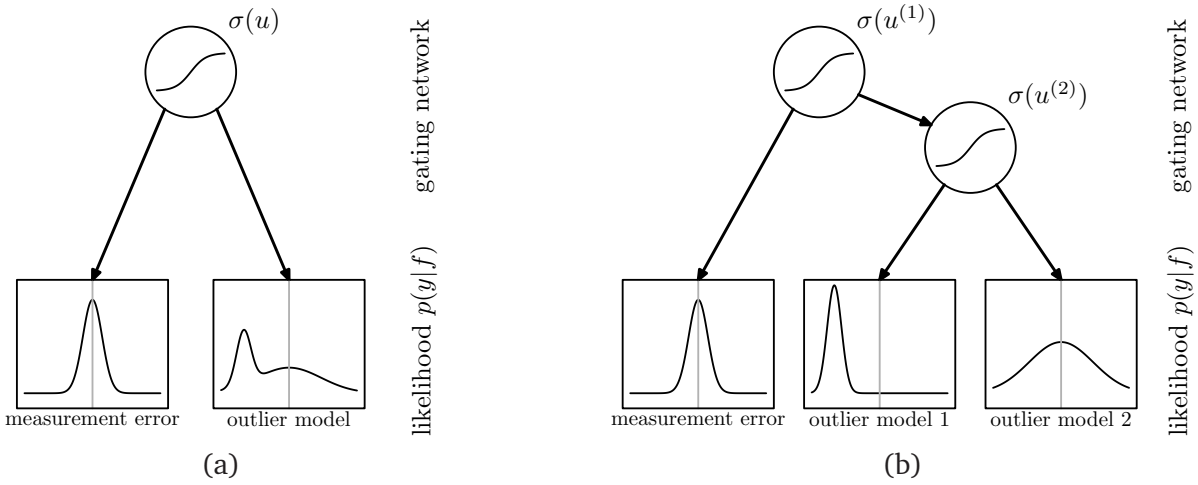


Figure 4.7: Two proposed methods for extending the TGP noise model: in panel (a), outlying observations are drawn from a mixture of two Gaussians, while in panel (b), the outlier mixture is split into its components, whose input-dependent responsibilities are assigned by an extra node in the gating network.

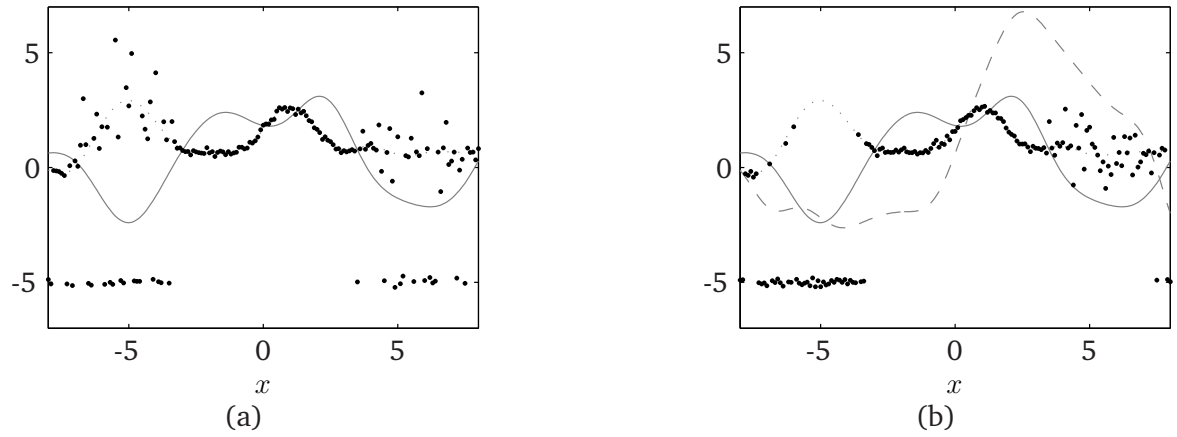


Figure 4.8: The black dots are "fantasy" data, drawn in panel (a) from the marginal likelihood for (4.6) and in panel (b) for (4.7). The solid grey line is the latent $\mathbf{u}^{(1)}$ process, which determines where outliers are likely to occur. The dashed grey line in the second panel is $\mathbf{u}^{(2)}$: if its value is less than zero, outlying observations tend to cluster around -5; where it is positive, a wide Gaussian corruption predominates.

as before, but now those clusters consist of independent samples from the mixture (4.6). A related problem is that, although we can learn the hyperparameters $\boldsymbol{\alpha}$ that maximize the evidence, there is always the constraint that the relative weight of outlier corruptions is constant across the entire domain: if the various models of corruption are themselves of interest, it is difficult to determine how their responsibilities vary within the data. A computational issue arises too: multimodality of the likelihood makes a multimodal posterior more likely than simply using a heavy-tailed model, and this can further hamper the convergence of EP.

For a richer alternative, the gating network can be extended to operate on a series of process variables $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(C)}$, in conjunction with an appropriate broadening of the sampling distribution. In the related mixture model of Tresp (2001), the exponential softmax function is used in the gating network, but to permit our continued use of EP in the tractable approximation of the posterior distribution, we use here a binary tree arrangement of cumulative Gaussian sigmoid functions. An example appears in fig. 4.7b, corresponding to a noise model with $C = 2$ and

$$\begin{aligned} p(y_n|f_n, u_n^{(1)}, u_n^{(2)}, f_0, \boldsymbol{\sigma}^2) = \sigma(u_n^{(1)})\mathcal{N}\big(y_n\,;\, f_n\,,\, \sigma_R^2\big) + \\ \sigma(-u_n^{(1)})\sigma(u_n^{(2)})\mathcal{N}\big(y_n\,;\, f_0\,,\, \sigma_{O_1}^2\big) + \\ \sigma(-u_n^{(1)})\sigma(-u_n^{(2)})\mathcal{N}\big(y_n\,;\, f_n\,,\, \sigma_{O_2}^2\big)\,. \end{aligned} \quad (4.7)$$

Typical data drawn from the marginal likelihoods of these two models are shown in fig. 4.8. In the following sections, we concentrate exclusively on the latter since it is considerably more flexible, and includes the simpler as a special case.

Evidently, we can further generalize (4.7) in a similar way to our development of the "ignorance" model into a full secondary process on $\mathbf{g}$. For example, we might introduce three GPs on latent $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{h}$, as well as the gating processes on $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$, arranging the former processes at the leaves of a tree constructed from the latter:

$$\begin{aligned} p(y_n|f_n, g_n, h_n, u_n^{(1)}, u_n^{(2)}, \boldsymbol{\sigma}^2) = \sigma(u_n^{(1)})\mathcal{N}\big(y_n\,;\, f_n\,,\, \sigma_f^2\big) + \\ \sigma(-u_n^{(1)})\sigma(u_n^{(2)})\mathcal{N}\big(y_n\,;\, g_n\,,\, \sigma_g^2\big) + \\ \sigma(-u_n^{(1)})\sigma(-u_n^{(2)})\mathcal{N}\big(y_n\,;\, h_n\,,\, \sigma_h^2\big)\,. \end{aligned} \quad (4.8)$$

There is theoretically no limit to how many processes we can use, but due to their interaction in the posterior, costs scale with the cube of this number, probably imposing

a fairly small practical limit.

Observe a further subtlety: since the gate on $\sigma(\mathbf{u}^{(1)})$ precedes that on $\sigma(\mathbf{u}^{(2)})$, we find the model on $\mathbf{f}$ takes precedence over models on $\mathbf{g}$ and $\mathbf{h}$. In consequence, there is no longer the symmetry between $\mathbf{f}$ and $\mathbf{g}$ of the two-component solution. This is reasonable if the extra distributions are describing only outliers, but may be less appropriate if they model equally valid extra processes in the data. In the more general mixture of Tresp (2000), the gates were not arranged in a tree but selection was via a softmax exponential. This maintains symmetry in the various latent signals, but cannot readily be allied with our deterministic inference procedure due to the interactions induced by the renormalization across gating processes.

4.5.2 Inference

Our extension of the TGP largely overcomes the problems encountered when a simple mixture of Gaussians (4.6) is used for the outlier distribution, but suffers a greater computational burden through the maintenance of a posterior distribution over $\mathbf{f}$ and all C $\mathbf{u}$s. Let this posterior approximation be

$$\mathcal{N}\left(\begin{bmatrix} \{\mathbf{u}^{(c)}\} \\ \mathbf{f} \end{bmatrix}; \mathbf{h}, \mathbf{A}\right), \qquad \text{where} \qquad \{\mathbf{u}^{(c)}\} = \begin{bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(C)} \end{bmatrix}.$$

We initialize

$$\mathbf{h} = \begin{bmatrix} \{\mathbf{m}^{(c)}\} \\ \mathbf{0} \end{bmatrix} \qquad \text{and} \qquad \mathbf{A} = \begin{bmatrix} \mathbf{K}_{\mathbf{u}^{(1)}\mathbf{u}^{(1)}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{\mathbf{u}^{(C)}\mathbf{u}^{(C)}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{K}_{\mathbf{ff}} \end{bmatrix},$$

with the natural generalization (to include $\mathbf{K}_{\mathbf{gg}}$) if we are using model (4.8). The non-zero means $\mathbf{m}^{(c)}$ for the $\mathbf{u}$-processes allow the adjustment of the frequency and nature of corruptions to fit our prior beliefs.

The covariance $\mathbf{A}$ is initially block diagonal: each block of size $N \times N$ consists of one of the kernel matrices $\mathbf{K}_{\mathbf{u}^{(c)}\mathbf{u}^{(c)}}$, or $\mathbf{K}_{\mathbf{ff}}$. We again use EP to refine the Gaussian approximation, and the algorithm of section 3.2.2 can be applied with minor changes. Primary

among these is how to estimate the marginal moments of the tilted distribution. The general expression for which we require a closed form is intractable:

$$Z = \int \cdots \int \sigma(u^{(1)}) \cdots \sigma(u^{(C)}) \mathcal{N}\big(f\,;\,\mu_f\,,\,\sigma_f^2\big)\,\mathcal{N}\left(\left[\begin{array}{c}\{u^{(c)}\}\\ f\end{array}\right]\,;\,\boldsymbol{\mu}^{\backslash n}\,,\,\boldsymbol{\Sigma}^{\backslash n}\right)\mathrm{d}\mathbf{u}\mathrm{d}f, \quad (4.9)$$

but again recognised as a close relation of the marginal likelihood for a GP classification model, with a data set of size C and with an extra Gaussian site at f. Using an auxiliary EP loop in the spirit of section 4.1, necessary derivatives $\boldsymbol{\alpha}$ and $\boldsymbol{\nu}$ of the partition function Z can be obtained by following the procedure of (4.2) and (4.3). In this inner loop, there are up to $C+1$ sites, although the number will depend on the arrangement of the gating network: we make a separate moment calculation for each *leaf* in the tree, and adding to that at f, each node on the path to the leaf contributes its own site approximation. Thus, when the path contains a single node the moment calculation is analytic, as exploited by the TGP, while longer paths call for the alternative approach. Although it is possible to incorporate the Gaussian over f directly into the "prior" of (4.9), it can equivalently be included in the EP iterations as an exact special case, for which we find

$$Z^{(f)} = \int \mathcal{N}\big(f\,;\,\mu_f\,,\,\sigma_f^2\big)\,\mathcal{N}\Big(f\,;\,\mu_f^{\backslash n}\,,\,\Sigma_f^{\backslash n}\Big)\,\mathrm{d}f = \mathcal{N}\Big(\mu_f\,;\,\mu_f^{\backslash n}\,,\,\Sigma_f^{\backslash n} + \sigma_f^2\Big)$$

The other sites are treated as for conventional GP classification, the relevant moments for which are reviewed in appendix B.

Returning to the main problem, we find it is now very similar to those we have already seen: there are N multivariate Gaussian site approximations, each of $C+1$ dimensions. Once we have the necessary moments, the mathematics of their refinement is essentially unchanged from the TGP case, except with rank-$(C+1)$ updates of the covariance at each iteration.

4.5.3 Motorcycle revisited

Let us return to Silverman's motorcycle set, which we encountered in the context of the TGP model in section 3.4.3. Our conclusion was that a two-component model was insufficient to capture the variance in the data, and with reference to the experiments of Rasmussen and Ghahramani (2002) that at least three components should be required.

To investigate whether an extra component may allow the TGP enough flexibility to fit the data well, we trained an extended twinned GP model—a "triplet" GP—using two $\mathbf{u}$ processes and a single mean $\mathbf{f}$ about which all noise is Gaussian distributed. Our results appear in fig. 4.9. There are now three noise models, and our experiments confirm that suitable variances can be learned for each, together with an appropriate gating function. For approximately $t < 13$ (where t is the index variable), the low-variance component is weighted heavily by $\mathbf{u}^{(1)}$, but this is almost entirely "switched off" for larger t. Beyond $t > 13$, $\mathbf{u}^{(2)}$ switches between a large-variance component in the central region, and a component of intermediate variance for $t > 42$. It can be observed that this fit is far superior to that of the TGP (fig. 3.11d); there is a much sharper transition between the low- and high-variance regions, and a better fit to the reduced variance data on the right-hand side. We also notice a spike in the mixing processes at around $t = 22$ which is not anomalous: the model is attempting to "pinch" the variance at the data (which are less spread at this point) but the global lengthscales are such that the $\mathbf{u}$ signals cannot grow large enough to make the effect visible.

Finally, it was discovered that for these highly heteroscedastic data, the EP iterations converged more reliably for the triplet model (that is, with less damping) than the earlier TGP, despite the increased complexity imparted by an additional process and noise model. We suspect that much of the instability in the TGP inference was caused by the model mismatch yielding a poorly-peaked posterior, a situation which is improved by the additional flexibility enjoyed in this case.

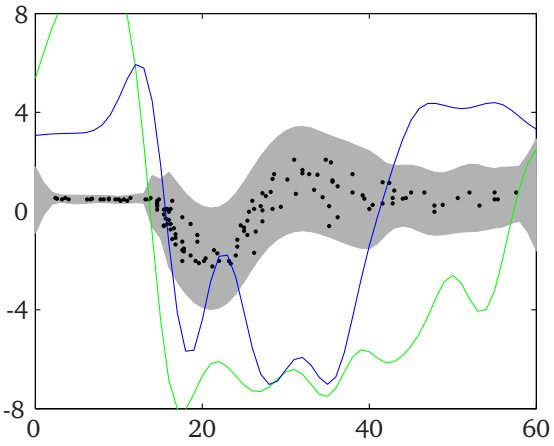


Figure 4.9: The green line is the mean process on $\mathbf{u}^{(1)}$, switching between the "real" component and the two outliers. The blue line is the mean of $\mathbf{u}^{(2)}$; when positive it preferentially chooses a medium-variance noise distribution, if it is negative the large-variance component is favoured. Compare with fig. 3.11d.

CHAPTER 5

Conclusions

THIS THESIS HAS presented methods for speeding up inference in Gaussian processes, for making them more robust to outlying observations, and for their efficient use in general mixture modelling tasks.

To summarize its contributions, in chapter 2 it was shown how to extend the pseudo-input regression model of Snelson and Ghahramani (2006a) to arbitrary likelihoods, using EP to drive the inference. In this model, the speed of training and prediction can be regulated a priori by controlling the size of the active set; furthermore, elements of this set become explicit hyperparameters of the kernel, and it was described how these could be learned as part of a continuous optimization. The experimental section was devoted to the task of binary classification, and in a detailed comparison on data sets of intermediate size, the sparse GP classifier was found to be highly competitive with other state of the art sparse methods. Theoretical and practical insight was provided on the problem of finding sparse solutions, and it was suggested that the very sparsest may only be found by global optimization rather than in a greedy manner. An investigation into the ability of a supervised dimensionality reduction technique to afford efficiency gains gave disappointing results, although its successful application in Snelson and Ghahramani (2006b) for regression models suggests this is may be as much a difficulty with the classification domain as with the concept.

Chapter 3 introduced a new robust model for regression. It was shown how inference could be conducted by EP with only a constant factor more complexity than for the standard Gaussian mixture which it generalizes. Experimentally, the new noise model was found to give more confident predictions on homoscedastic data, and to remain more stable with respect to clusters of outlying observations than the mixture. Also observed was improved EP convergence, for which practical measures were suggested to aid the process. Some ability to model heteroscedastic data was exhibited, and the sources of difficulty were discussed in detail together with possible resolutions.

In chapter 4 the twinned GP was further developed, yielding new models for robust classification, regression and mixture modelling. In all these cases, it was shown how inference could be conducted without recourse to costly Monte Carlo integration; the EP framework was used throughout, and in a new manner which employed EP subroutines to evaluate intractable site moments. By adding a third component to the TGP model from chapter 3, the heteroscedastic data that was modelled inadequately there could be fit with significantly improved predictive density.

We identify several areas for further research. Our work with the generalized sparse pseudo-input GP is restricted to probit noise, but it would be interesting to extend the model to other regimes; for example, Seeger et al. (2006) discuss ordered regression and multi-class classification for the IVM. They also consider “virtual informative vectors”, in which the active set is augmented with extra inputs derived by known invariant deformations such as rotation and translation of images. The technique was found to improve the performance of the IVM and SVM (where it originated), and its effect on our model would constitute a fruitful investigation.

The twinned GP opens numerous avenues of inquiry. We are particularly curious to explore an alternative inference procedure based on variational methods. Since they tend to focus on a single mode of the posterior, it is interesting to speculate how a problem of clustered outliers would be resolved (see fig. 3.12). Our primary motivation is that the issue of EP convergence remained in some cases problematic. An alternative solution would be to employ more expensive “double loop” algorithms (Opper and Winther, 2005) which are assured to find a fixed point of the associated energy function. Other valuable extensions include a sparsified model, bringing the ideas of chapters 2 and 3 together, and more elaborate structures (we note, for example, that the TGP may be combined usefully with the ordinal regression model described in Paquet et al. (2005)).

APPENDIX A

Mathematical preliminaries

A.1 Exponential families

A set of distributions $\mathcal{F}$ with densities

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp\left(\boldsymbol{\theta}^T\boldsymbol{\tau}(\mathbf{x}) - \boldsymbol{\Phi}(\boldsymbol{\theta})\right), \text{ where } \boldsymbol{\theta} \in \Theta,$$
$$\boldsymbol{\Phi}(\boldsymbol{\theta}) = \log \int \exp\left(\boldsymbol{\theta}^T\boldsymbol{\tau}(\mathbf{x})\right) \mathrm{d}\mu(\mathbf{x})$$

w.r.t. a base measure μ is called an exponential family. The *natural parameter space* is Θ, $\boldsymbol{\theta}$ the *natural parameters*, $\boldsymbol{\tau}(\mathbf{x})$ the *sufficient statistics*, and $\boldsymbol{\Phi}$ the log partition function. The *moment parameters* are $\boldsymbol{\eta} = \mathbb{E}[\boldsymbol{\tau}(\mathbf{x})]$. Many familiar distributions are exponential family; of particular interest will be the Gaussian, for which

$$\boldsymbol{\tau}(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}\mathbf{x}^T \end{bmatrix}, \qquad \boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ -\frac{1}{2}\boldsymbol{\Sigma}^{-1} \end{bmatrix}, \qquad \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T \end{bmatrix}. \tag{A.1}$$

A.2 The Gaussian distribution

The set of Gaussian distributions is closed under many common operations. In particular, marginal and conditional distributions are also Gaussian; that is, if

$$p(\mathbf{x},\mathbf{y}) = \mathcal{N}\left(\left[\begin{array}{c}\mathbf{x}\\ \mathbf{y}\end{array}\right]; \left[\begin{array}{c}\mathbf{a}\\ \mathbf{b}\end{array}\right], \left[\begin{array}{cc}\mathbf{A} & \mathbf{C}\\ \mathbf{C}^T & \mathbf{B}\end{array}\right]\right),$$

$$\text{then } \int p(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{y} = \mathcal{N}(\mathbf{x}\,;\,\mathbf{a}\,,\,\mathbf{A})\,;$$

$$\text{and } p(\mathbf{x}|\mathbf{y}) = \mathcal{N}\big(\mathbf{x}\,;\,\mathbf{a}+\mathbf{C}^T\mathbf{B}^{-1}(\mathbf{y}-\mathbf{b})\,,\,\mathbf{A}-\mathbf{C}^T\mathbf{B}^{-1}\mathbf{C}\big)\,.$$

Furthermore, the product of two Gaussians is proportional to a third Gaussian, and the proportionality is regulated by a fourth. This result holds in general for the product under linear projection $\mathbf{P}$:

$$\mathcal{N}(\mathbf{x}\,;\,\mathbf{a}\,,\,\mathbf{A})\,\mathcal{N}(\mathbf{P}\mathbf{x}\,;\,\mathbf{b}\,,\,\mathbf{B}) = \mathcal{N}\big(\mathbf{b}\,;\,\mathbf{P}\mathbf{a}\,,\,\mathbf{B}+\mathbf{P}\mathbf{A}\mathbf{P}^T\big)\,\mathcal{N}(\mathbf{x}\,;\,\mathbf{c}\,,\,\mathbf{C})\,,$$

where $\mathbf{C}^{-1} = \mathbf{A}^{-1}+\mathbf{P}^T\mathbf{B}^{-1}\mathbf{P}$, and $\mathbf{c} = \mathbf{C}\left(\mathbf{A}^{-1}\mathbf{a}+\mathbf{P}^T\mathbf{B}^{-1}\mathbf{b}\right)$. The marginal follows trivially:

$$\int \mathcal{N}(\mathbf{x}\,;\,\mathbf{a}\,,\,\mathbf{A})\,\mathcal{N}(\mathbf{P}\mathbf{x}\,;\,\mathbf{b}\,,\,\mathbf{B})\,\mathrm{d}\mathbf{x} = \mathcal{N}\big(\mathbf{b}\,;\,\mathbf{P}\mathbf{a}\,,\,\mathbf{B}+\mathbf{P}\mathbf{A}\mathbf{P}^T\big)\,.$$

A.2.1 Derivatives of Gaussian forms

It is helpful to remember the following identities, in which $\sigma(u) = \int_{-\infty}^{u}\mathcal{N}(z\,;\,0\,,\,1)\,\mathrm{d}z$, the probit or cumulative distribution function of the Gaussian.

$$\begin{aligned}\frac{\partial\mathcal{N}(\mathbf{r}\,;\,\mathbf{m}\,,\,\mathbf{\Sigma})}{\partial\mathbf{m}} &= \mathcal{N}(\mathbf{r}\,;\,\mathbf{m}\,,\,\mathbf{\Sigma})\,\mathbf{\Sigma}^{-1}(\mathbf{r}-\mathbf{m});\\ \frac{\partial\mathcal{N}(u(\mathbf{z})\,;\,0\,,\,1)}{\partial\mathbf{z}} &= -u(\mathbf{z})\mathcal{N}(u(\mathbf{z})\,;\,0\,,\,1)\,\frac{\partial u(\mathbf{z})}{\partial\mathbf{z}};\\ \frac{\partial\sigma(u(\mathbf{z}))}{\partial\mathbf{z}} &= \mathcal{N}(u(\mathbf{z})\,;\,0\,,\,1)\,\frac{\partial u(\mathbf{z})}{\partial\mathbf{z}}.\end{aligned}$$

A.3 Matrix algebra

The matrix inversion lemma, or the Sherman-Morrison-Woodbury formula, states that

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

and equivalently for determinants,

$$\log|\mathbf{A}+\mathbf{PBP}^T| = \log|\mathbf{A}| + \log|\mathbf{B}| + \log|\mathbf{B}^{-1}+\mathbf{P}^T\mathbf{A}^{-1}\mathbf{P}|,$$

where $\mathbf{A}$ is $N \times N$, $\mathbf{B}$ is $M \times M$, and $\mathbf{P}$ is $N \times M$.

A.3.1 Cholesky decomposition

The Cholesky factorization of positive definite matrices should always be preferred to generalized inversion: it is more stable, slightly more efficient, and the triangular factors can be used to calculate common forms. They appear in two orientations, upper $\mathbf{R}$ and lower $\mathbf{L}$:

$$\mathbf{K} = \mathbf{L}\mathbf{L}^T = \mathbf{R}^T\mathbf{R},$$

from which quadratic terms can be calculated in $\mathcal{O}(N^2)$ by backsubstitution:

$$\mathbf{v}^T\mathbf{K}^{-1}\mathbf{v} = \mathbf{v}^T\mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{v} = \|\mathbf{L}\backslash\mathbf{v}\|^2.$$

Determinants are also readily evaluated:

$$\log\det\mathbf{K} = 2\sum_n \log(L_{nn})$$

A.3.2 Derivatives of matrix forms

In the following, $\bullet$ denotes the element-wise or Hadamard product:

$$\frac{\partial\mathbf{K}^{-1}}{\partial\theta} = -\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta}\mathbf{K}^{-1} \tag{A.3}$$

$$\frac{\partial\log|\mathbf{K}(\theta)|}{\partial\theta} = \mathsf{tr}\left(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta}\right) = \sum_{m,n}\left(\mathbf{K}^{-1}\bullet\frac{\partial\mathbf{K}}{\partial\theta}\right)_{m,n}. \tag{A.4}$$

A.4 Kullback-Leibler divergence

The asymmetric KL-divergence between two probability distributions is

$$\mathsf{KL}\left(p(\mathbf{x})\|q(\mathbf{x})\right) = \int p(\mathbf{x})\ln\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right)\mathrm{d}\mathbf{x}.$$

APPENDIX B

Sparse Gaussian process classification

B.1 EP for Gaussian process classification

We make use of the EP framework provided in section 1.3. The site functions are evaluations of the probit with bias b, $t_n(f_n) = p(y_n|f_n) = \sigma(y_n(f_n + b))$, so that the zeroth moment of the tilted distribution is

$$\begin{aligned} Z_n &= \int \sigma(y_n(f_n+b))\mathcal{N}\left(f_n\,;\,\mu_{\backslash n}\,,\,\sigma^2_{\backslash n}\right)\mathrm{d}f_n \\ &= \int\int_{z_n=-\infty}^{y_n(f_n+b)} \mathcal{N}(z_n\,;\,0\,,\,1)\,\mathcal{N}\left(f_n\,;\,\mu_{\backslash n}\,,\,\sigma^2_{\backslash n}\right)\mathrm{d}z_n\mathrm{d}f_n \\ &= \int_{z_n=0}^{\infty}\int \mathcal{N}(y_n(f_n+b)\,;\,z_n\,,\,1)\,\mathcal{N}\left(f_n\,;\,\mu_{\backslash n}\,,\,\sigma^2_{\backslash n}\right)\mathrm{d}f_n\mathrm{d}z_n \\ &= \int_{z_n=0}^{\infty} \mathcal{N}\left(z_n\,;\,y_n(\mu_{\backslash n}+b)\,,\,1+\sigma^2_{\backslash n}\right)\mathrm{d}z_n = \sigma\left(\frac{y_n(\mu_{\backslash n}+b)}{\sqrt{1+\sigma^2_{\backslash n}}}\right), \end{aligned}$$

whose derivatives with respect to μ are

$$\frac{\partial Z_n}{\partial \mu_{\backslash n}} = \frac{y_n}{\sqrt{1+\sigma^2_{\backslash n}}}\mathcal{N}\left(\frac{y_n(\mu_{\backslash n}+b)}{\sqrt{1+\sigma^2_{\backslash n}}}\right); \quad \frac{\partial^2 Z_n}{\partial \mu^2_{\backslash n}} = -\frac{y_n(\mu_{\backslash n}+b)}{\left(1+\sigma^2_{\backslash n}\right)^{3/2}}\mathcal{N}\left(\frac{y_n(\mu_{\backslash n}+b)}{\sqrt{1+\sigma^2_{\backslash n}}}\right),$$

so that

$$\alpha_n = \frac{y_n}{Z_n\sqrt{1+\sigma^2_{\backslash n}}}\mathcal{N}\left(\frac{y_n(\mu_{\backslash n}+b)}{\sqrt{1+\sigma^2_{\backslash n}}}\right); \qquad \nu_n = \alpha_n\left(\alpha_n + \frac{\mu_{\backslash n}+b}{1+\sigma^2_{\backslash n}}\right).$$

These expressions can be plugged into the update rules (1.13) and iterated to yield the EP approximation to the posterior.

B.2 Model selection for the generalized FITC approximation

Model selection is complicated by two issues: first, we must consider gradients of the kernel with respect not only to its hyperparameters (such as lengthscale and amplitude), but also to locations of the pseudo-inputs $\bar{\mathbf{X}}$. Second, we must be vigilant that the complexity of the derivative calculations remains bounded by $\mathcal{O}(NM^2)$, which requires expanding $N\times N$ matrices into their components, typically the sum of a diagonal term and a rank-M term.

We have from section 1.3.2 that $\nabla_{\boldsymbol{\xi}^{(0)}} L = \boldsymbol{\eta} - \boldsymbol{\eta}^{(0)}$, where L is the log marginal likelihood, $\boldsymbol{\xi}^{(0)}$ are the natural parameters of the prior, and $\boldsymbol{\eta}$ are the moment parameters. In the prior,

$$\boldsymbol{\mu} = \mathbf{0} \qquad \text{and} \qquad \boldsymbol{\Sigma} = \mathsf{diag}\left(\mathbf{K}_{\mathbf{ff}} - \mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}\right) + \mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}.$$

Write $\mathbf{D}_0$ for the diagonal term in $\boldsymbol{\Sigma}$, and write the posterior covariance in terms site precisions $\boldsymbol{\Pi}$ as $(\boldsymbol{\Sigma}^{-1}+\boldsymbol{\Pi})^{-1} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}+\boldsymbol{\Pi}^{-1}\right)^{-1}\boldsymbol{\Sigma}$. Then

$$\begin{aligned}\nabla_{\boldsymbol{\xi}^{(0)}} L &= -\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}+\boldsymbol{\Pi}^{-1}\right)^{-1}\boldsymbol{\Sigma} + \left(\boldsymbol{\Sigma}\mathbf{b} - \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}+\boldsymbol{\Pi}^{-1}\right)^{-1}\boldsymbol{\Sigma}\mathbf{b}\right)\left(\boldsymbol{\Sigma}\mathbf{b} - \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}+\boldsymbol{\Pi}^{-1}\right)^{-1}\boldsymbol{\Sigma}\mathbf{b}\right)^T \\ &= -\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}+\boldsymbol{\Pi}^{-1}\right)^{-1}\boldsymbol{\Sigma} + \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}+\boldsymbol{\Pi}^{-1}\right)^{-1}\boldsymbol{\Pi}^{-1}\mathbf{b}\mathbf{b}^T\boldsymbol{\Pi}^{-1}\left(\boldsymbol{\Sigma}+\boldsymbol{\Pi}^{-1}\right)^{-1}\boldsymbol{\Sigma},\end{aligned}$$

where $\mathbf{b}$ are site parameters corresponding to precision times mean. This leaves the calculation of $\nabla_{\boldsymbol{\theta}}\boldsymbol{\xi}^{(0)}$ for some $\boldsymbol{\theta}$ (either a kernel hyperparameter or the coordinate of a pseudo-input). In the latter case, we find we can combine derivative calculations for all dimensions while still dealing with standard matrices (rather than higher-order tensors) because moving the jth pseudo-input affects only covariance calculations that involve $\bar{\mathbf{x}}_j$: most entries in the tensor evaluate to zero and can be ignored. Recalling

(A.3), and dropping $\boldsymbol{\theta}$ subscripts on ∇,

$$\begin{aligned}\nabla L &= \mathsf{Tr}\left(\left(\nabla \boldsymbol{\xi}^{(0)}\right)\left(\nabla_{\boldsymbol{\xi}^{(0)}} L\right)\right) \\ &= \frac{1}{2}\mathsf{Tr}\left(\left(\boldsymbol{\Sigma}+\boldsymbol{\Pi}^{-1}\right)^{-1}(\nabla\boldsymbol{\Sigma})\right) - \frac{1}{2}\mathbf{b}^T\boldsymbol{\Pi}^{-1}\left(\boldsymbol{\Sigma}+\boldsymbol{\Pi}^{-1}\right)^{-1}(\nabla\boldsymbol{\Sigma})\left(\boldsymbol{\Sigma}+\boldsymbol{\Pi}^{-1}\right)^{-1}\boldsymbol{\Pi}^{-1}\mathbf{b}.\end{aligned} \tag{B.1}$$

Now let $\mathbf{E} = \mathbf{D}_0 + \boldsymbol{\Pi}^{-1}$, which is diagonal and allows easy inversion;

$$\begin{aligned}\left(\boldsymbol{\Sigma}+\boldsymbol{\Pi}^{-1}\right)^{-1} &= \mathbf{E}^{-1} - \mathbf{E}^{-1}\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\left(\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}\mathbf{E}^{-1}\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}} + \mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}\right)^{-1}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}\mathbf{E}^{-1} \\ &= \mathbf{E}^{-1} - \mathbf{B}^T\mathbf{B},\end{aligned} \tag{B.2}$$

where $\mathbf{B} = \mathsf{chol}\left(\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}\mathbf{E}^{-1}\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}} + \mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}\right)\backslash\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{E}^{-1}$ has dimensions $M \times N$. Derivatives $\nabla\boldsymbol{\Sigma}$ of the prior covariance with respect to hyperparameters of the kernel include the term

$$\nabla\mathbf{D}_0 = \mathsf{diag}\left(\nabla\mathbf{K}_{\mathbf{f}\mathbf{f}}\right) - 2\mathsf{diag}\left(\left(\nabla\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\right)\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}\right) + \mathsf{diag}\left(\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\left(\nabla\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}\right)\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}\right). \tag{B.3}$$

For stationary covariance functions, $\nabla\mathbf{K}_{\mathbf{f}\mathbf{f}}$ has constant diagonal; the other terms are evaluated in $\mathcal{O}(NM^2)$, and we should retain the partial derivative matrices they contain, since

$$\nabla\boldsymbol{\Sigma} = \nabla\mathbf{D}_0 + \left(\nabla\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\right)\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}} + \mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\left(\nabla\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}\right)\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}} + \mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}\left(\nabla\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}\right). \tag{B.4}$$

Multiplied together, (B.2) and (B.4) provide the necessary terms for evaluating (B.1) within the complexity bounds.

We turn now to derivatives $\nabla_{\bar{\mathbf{x}}_j} L$ of the log marginal likelihood with respect to points in the active set. It was observed how these are simplified by an independence property, namely that moving the dth component of $\bar{\mathbf{x}}_j$ cannot affect $\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}$ or $\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}$ at rows or columns not involving the jth pseudo-input, so that we can consider derivatives with respect to full $\bar{\mathbf{x}}_j$ while dealing only with standard matrices. We consider the various terms we need in turn, namely the derivatives of the diagonal and low-rank components of the covariance.

The tensor $\nabla_{\bar{\mathbf{x}}_j}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}$, which may be visualized as a cuboid $M \times N$ with depth D (where D is the dimensionality of the data), can be condensed for efficiency into a $D \times N$ matrix since only the jth row is non-zero. Similarly, $\nabla_{\bar{\mathbf{x}}_j}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}$ is zero everywhere except along the jth row and column; it can be condensed into a $D \times M$ matrix consist-

ing only of (half) these non-zero elements (the tensor must be symmetric). Finally, consider $\nabla_{\bar{\mathbf{x}}_j}\mathbf{D}_0$ (B.3): the term $\mathbf{K}_{\mathbf{ff}}$ has derivative zero, since it is independent of the pseudo-inputs altogether. The term $\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\left(\nabla_{\bar{\mathbf{x}}_j}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}\right)$ is an $N \times N \times D$ tensor, but we require only the diagonal which is $N \times D$. The final term is the diagonal of $\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\left(\nabla_{\bar{\mathbf{x}}_j}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}\right)\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}$; we make the same simplification, using the condensed $D \times M$ form of $\nabla_{\bar{\mathbf{x}}_j}\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}$ but remembering to account for both row and column of the original tensor.

B.3 Dimensionality reduction

In section 2.5, we suggested how the training data could be projected in a supervised manner onto a low-dimensional manifold in order to accelerate in many dimensions the learning process, or to allow a greater number of pseudo-inputs without raising its cost. Unfortunately, there is no easy way to deal with entire rows or columns of the projection matrix in the manner we could with the separate components of each pseudo-input, since there is no comparable independence property. Hence, we must iterate over every element P_{ij}, although only terms involving $\nabla_{P_{ij}}\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}$ survive:

$$\nabla\mathbf{D}_0 = -2\mathsf{diag}\left(\left(\nabla_{P_{ij}}\mathbf{K}_{\mathbf{f}\bar{\mathbf{f}}}\right)\mathbf{K}_{\bar{\mathbf{f}}\bar{\mathbf{f}}}^{-1}\mathbf{K}_{\bar{\mathbf{f}}\mathbf{f}}\right),$$

where we have assumed a stationary covariance to eliminate $\nabla_{P_{ij}}\mathbf{K}_{\mathbf{ff}}$.

APPENDIX C

Robust Gaussian process regression

C.1 Inference

Recall that the prior over $\mathbf{u}$ and $\mathbf{f}$ is

$$p\left(\left[\begin{array}{c}\mathbf{u}\\ \mathbf{f}\end{array}\right]\middle|\mathbf{X}\right)=\mathcal{N}\left(\left[\begin{array}{c}\mathbf{u}\\ \mathbf{f}\end{array}\right];\left[\begin{array}{c}\mathbf{m_u}\\ \mathbf{0}\end{array}\right],\left[\begin{array}{cc}\mathbf{K_{uu}} & \mathbf{0}\\ \mathbf{0} & \mathbf{K_{ff}}\end{array}\right]\right)$$

and the likelihood factorizes into a product of terms

$$t_n(y_n|f_n,u_n)=\sigma(u_n)\mathcal{N}\left(y_n\,;\,f_n\,,\,\sigma_R^2\right)+\sigma(-u_n)\mathcal{N}\left(y_n\,;\,f_n\,,\,\sigma_O^2\right).$$

We use N scaled natural Gaussian site functions $s_n\tilde{t}_n$ to construct an approximate posterior distribution

$$\mathcal{N}\left(\left[\begin{array}{c}\mathbf{u}\\ \mathbf{f}\end{array}\right];\left[\begin{array}{c}\mathbf{m_u}\\ \mathbf{0}\end{array}\right],\left[\begin{array}{cc}\mathbf{K_{uu}} & \mathbf{0}\\ \mathbf{0} & \mathbf{K_{ff}}\end{array}\right]\right)\prod_{n=1}^{N}s_n\tilde{t}_n\left(\left[\begin{array}{c}u_n\\ f_n\end{array}\right];\mathbf{b}_n\,,\,\mathbf{\Pi}_n\right).$$

We require the moments of the tilted distribution, the product of a likelihood term t_n and the cavity distribution $\mathcal{N}([u;f]\,;\,\boldsymbol{\mu}\,,\,\mathbf{\Sigma})$. Dropping the n subscripts, the zeroth

moment for the component corresponding to real data is

$$\begin{aligned}
Z^R &= \iint \sigma(u)\mathcal{N}\big(y\,;\,f\,,\,\sigma_R^2\big)\,\mathcal{N}\left(\begin{bmatrix} u \\ f \end{bmatrix};\,\boldsymbol{\mu}\,,\,\boldsymbol{\Sigma}\right)\mathrm{d}u\mathrm{d}f \\
&= \iiint_{z=-\infty}^{u} \mathcal{N}(z\,;\,0\,,\,1)\,\mathcal{N}\big(y\,;\,f\,,\,\sigma_R^2\big)\,\mathcal{N}\left(\begin{bmatrix} u \\ f \end{bmatrix};\,\boldsymbol{\mu}\,,\,\boldsymbol{\Sigma}\right)\mathrm{d}z\mathrm{d}u\mathrm{d}f \\
&= \int_{z=0}^{\infty}\iint \mathcal{N}(u\,;\,z\,,\,1)\,\mathcal{N}\big(f\,;\,y\,,\,\sigma_R^2\big)\,\mathcal{N}\left(\begin{bmatrix} u \\ f \end{bmatrix};\,\boldsymbol{\mu}\,,\,\boldsymbol{\Sigma}\right)\mathrm{d}u\mathrm{d}f\mathrm{d}z \\
&= \int_{z=0}^{\infty}\iint_{f} \mathcal{N}\left(\begin{bmatrix} u \\ f \end{bmatrix};\begin{bmatrix} z \\ y \end{bmatrix},\begin{bmatrix} 1 & 0 \\ 0 & \sigma_R^2 \end{bmatrix}\right)\mathcal{N}\left(\begin{bmatrix} u \\ f \end{bmatrix};\,\boldsymbol{\mu}\,,\,\boldsymbol{\Sigma}\right)\mathrm{d}u\mathrm{d}f\mathrm{d}z \\
&= \int_{z=0}^{\infty} \mathcal{N}\left(\begin{bmatrix} z \\ y \end{bmatrix};\,\boldsymbol{\mu}\,,\,\begin{bmatrix} 1 & 0 \\ 0 & \sigma_R^2 \end{bmatrix}+\boldsymbol{\Sigma}\right)\mathrm{d}z,
\end{aligned}$$

where the final marginalization is a standard result. If we write the inner Gaussian as

$$\mathcal{N}\left(\begin{bmatrix} z_n \\ y_n \end{bmatrix};\begin{bmatrix} \mu_u \\ \mu_f \end{bmatrix},\begin{bmatrix} A & C \\ C & B_R \end{bmatrix}\right),$$

then

$$\begin{aligned}
Z^R &= \mathcal{N}(y\,;\,\mu_f\,,\,B_R)\int_{z=0}^{\infty}\mathcal{N}\left(z\,;\,\mu_u+\frac{C(y-\mu_f)}{B_R}\,,\,A-\frac{C^2}{B_R}\right)\mathrm{d}z \\
&= \mathcal{N}(y\,;\,\mu_f\,,\,B_R)\,\sigma(q), \qquad \text{(C.1)}
\end{aligned}$$

where

$$q=\frac{\mu_u+\frac{C}{B_R}(y-\mu_f)}{\sqrt{A-\frac{C^2}{B_R}}}.$$

A little algebra yields analytic expressions for the partial derivatives

$$\frac{\partial Z^R}{\partial \mu_u} = \frac{\mathcal{N}(y\,;\,\mu_f\,,\,B_R)\,\mathcal{N}(q)}{\sqrt{A - \frac{C^2}{B_R}}}; \tag{C.2a}$$

$$\frac{\partial Z^R}{\partial \mu_f} = \frac{\mathcal{N}(y\,;\,\mu_f\,,\,B_R)}{B_R}\left((y-\mu_f)\sigma(q) - \frac{C\mathcal{N}(q)}{\sqrt{A - \frac{C^2}{B_R}}}\right); \tag{C.2b}$$

$$\frac{\partial^2 Z^R}{\partial \mu_u^2} = -\frac{q\mathcal{N}(q)\mathcal{N}(y\,;\,\mu_f\,,\,B_R)}{A - \frac{C^2}{B_R}}; \tag{C.2c}$$

$$\frac{\partial^2 Z^R}{\partial \mu_f^2} = \frac{\mathcal{N}(y\,;\,\mu_f\,,\,B_R)}{B_R^2}\left(\sigma(q)\big((y-\mu_f)^2 - B_R\big) - \frac{C\mathcal{N}(q)}{\sqrt{A - \frac{C^2}{B_R}}}\left(2(y-\mu_f) + \frac{qC}{\sqrt{A - \frac{C^2}{B_R}}}\right)\right); \tag{C.2d}$$

$$\frac{\partial^2 Z^R}{\partial \mu_u \partial \mu_f} = \frac{\mathcal{N}(q)\mathcal{N}(y\,;\,\mu_f\,,\,B_R)}{B_R\sqrt{A - \frac{C^2}{B_R}}}\left(y - \mu_f + \frac{qC}{\sqrt{A - \frac{C^2}{B_R}}}\right) = \frac{\partial^2 Z^R}{\partial \mu_f \partial \mu_u}. \tag{C.2e}$$

The integral for the outlier component develops in a similar way:

$$Z^O = \mathcal{N}(y\,;\,\mu_f\,,\,B_O)\,\sigma(-q), \tag{C.3}$$

hence

$$\frac{\partial Z^O}{\partial \mu_u} = -\frac{\mathcal{N}(y\,;\,\mu_f\,,\,B_O)\,\mathcal{N}(q)}{\sqrt{A - \frac{C^2}{B_O}}}; \tag{C.4a}$$

$$\frac{\partial Z^O}{\partial \mu_f} = \frac{\mathcal{N}(y\,;\,\mu_f\,,\,B_O)}{B_O}\left((y-\mu_f)\sigma(q) + \frac{C\mathcal{N}(q)}{\sqrt{A - \frac{C^2}{B_O}}}\right); \tag{C.4b}$$

$$\frac{\partial^2 Z^O}{\partial \mu_u^2} = \frac{q\mathcal{N}(q)\mathcal{N}(y\,;\,\mu_f\,,\,B)}{A - \frac{C^2}{B}}; \tag{C.4c}$$

$$\frac{\partial^2 Z^O}{\partial \mu_f^2} = \frac{\mathcal{N}(y\,;\,\mu_f\,,\,B_O)}{B_O^2}\left(\sigma(-q)\big((y-\mu_f)^2 - B_O\big) - \frac{C\mathcal{N}(q)}{\sqrt{A - \frac{C^2}{B_O}}}\left(2(y-\mu_f) + \frac{qC}{\sqrt{A - \frac{C^2}{B_O}}}\right)\right); \tag{C.4d}$$

$$\frac{\partial^2 Z^O}{\partial \mu_u \partial \mu_f} = -\frac{\mathcal{N}(q)\mathcal{N}(y_n\,;\,\mu_f\,,\,B_O)}{B_O\sqrt{A - \frac{C^2}{B_O}}}\left(y - \mu_f + \frac{qC}{\sqrt{A - \frac{C^2}{B_O}}}\right) = \frac{\partial^2 Z^O}{\partial \mu_f \partial \mu_u}. \tag{C.4e}$$

C.2 Predictions

In the following, we partition the full posterior $\mathcal{N}\left(\begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}; \begin{bmatrix}\mathbf{h_u}\\ \mathbf{h_f}\end{bmatrix}, \begin{bmatrix}\mathbf{A_{uu}} & \mathbf{A_{uf}}\\ \mathbf{A_{fu}} & \mathbf{A_{ff}}\end{bmatrix}\right)$.

If the outlier component describes only nuisance noise that should be eliminated, we require at test inputs $\mathbf{x}_\star$ only the marginal distribution $p(f_\star|\mathbf{x}_\star, \mathbf{X}, \mathbf{y})$ obtained by marginalizing $\mathbf{u}$:

$$\begin{aligned} p(\mathbf{f}_\star|\mathbf{X}_\star, X, \mathbf{y}) &= \int p(\mathbf{f}_\star|\mathbf{X}_\star, \mathbf{f}) p(\mathbf{f}|\mathbf{X}, \mathbf{y}) \mathrm{df} \\ &\approx \int \mathcal{N}\left(\mathbf{f}_\star\,;\, \mathbf{k}_{\mathbf{f}\star}^T \mathbf{K}_{\mathbf{ff}}^{-1}\mathbf{f}\,,\, \mathbf{k}_{\star\star}^{(f)} - \mathbf{k}_{\mathbf{f}\star}^T \mathbf{K}_{\mathbf{ff}}^{-1}\mathbf{k}_{\mathbf{f}\star}\right) \mathcal{N}(\mathbf{f}\,;\, \mathbf{h_f}\,,\, \mathbf{A_{ff}})\, \mathrm{df} \\ &= \mathcal{N}\left(\mathbf{f}_\star\,;\, \mathbf{k}_{\mathbf{f}\star}^T \mathbf{K}_{\mathbf{ff}}^{-1}\mathbf{h_f}\,,\, \mathbf{k}_{\star\star}^{(f)} - \mathbf{k}_{\mathbf{f}\star}^T \mathbf{K}_{\mathbf{ff}}^{-1}\left(\mathbf{K_{ff}} - \mathbf{A_{ff}}\right)\mathbf{K}_{\mathbf{ff}}^{-1}\mathbf{k}_{\mathbf{f}\star}\right). \end{aligned}$$

Alternatively, the noise process may be of interest, in which case we require predictions

$$\begin{aligned} p(y_\star|\mathbf{x}_\star, \mathbf{X}, \mathbf{y}) &= \iint p\left(y_\star \middle| \mathbf{x}_\star, \begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}\right) p\left(\begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix} \middle| \mathbf{X}, \mathbf{y}\right) \mathrm{dudf} \\ &\approx \iiiint p\left(y_\star \middle| \mathbf{x}_\star, \begin{bmatrix}u_\star\\ f_\star\end{bmatrix}\right) p\left(\begin{bmatrix}u_\star\\ f_\star\end{bmatrix} \middle| \begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}\right) \mathcal{N}\left(\begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}; \mathbf{h}\,,\, \mathbf{A}\right) \mathrm{d}u_\star \mathrm{d}f_\star \mathrm{dudf}. \end{aligned}$$

The first term is the likelihood, while the second is a conditional Gaussian

$$p\left(\begin{bmatrix}u_\star\\ f_\star\end{bmatrix} \middle| \begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}u_\star\\ f_\star\end{bmatrix}; \begin{bmatrix}P & \mathbf{0}\\ \mathbf{0} & Q\end{bmatrix}^T \begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}, \begin{bmatrix}R & 0\\ 0 & S\end{bmatrix}\right),$$

where

$$P^T = \mathbf{k}_{u\star}^T \mathbf{K}_{\mathbf{uu}}^{-1}; \quad Q^T = \mathbf{k}_{f\star}^T \mathbf{K}_{\mathbf{ff}}^{-1}; \quad R = k_{\star\star}^{(u)} - \mathbf{k}_{u\star}^T \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{u\star}; \quad S = k_{\star\star}^{(f)} - \mathbf{k}_{f\star}^T \mathbf{K}_{\mathbf{ff}}^{-1}\mathbf{k}_{f\star}.$$

If we first marginalize the final two terms of the integrand

$$\begin{aligned} &\iint \mathcal{N}\left(\begin{bmatrix}u_\star\\ f_\star\end{bmatrix}; \begin{bmatrix}P & \mathbf{0}\\ \mathbf{0} & Q\end{bmatrix}^T \begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}, \begin{bmatrix}R & 0\\ 0 & S\end{bmatrix}\right) \mathcal{N}\left(\begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}; \mathbf{h}\,,\, \mathbf{A}\right) \mathrm{dudf} \\ &= \mathcal{N}\left(\begin{bmatrix}u_\star\\ f_\star\end{bmatrix}; \begin{bmatrix}P & \mathbf{0}\\ \mathbf{0} & Q\end{bmatrix}^T \mathbf{h}\,,\, \begin{bmatrix}R & 0\\ 0 & S\end{bmatrix} + \begin{bmatrix}P & \mathbf{0}\\ \mathbf{0} & Q\end{bmatrix}^T \mathbf{A} \begin{bmatrix}P & \mathbf{0}\\ \mathbf{0} & Q\end{bmatrix}\right) \doteq \mathcal{N}\left(\begin{bmatrix}u_\star\\ f_\star\end{bmatrix}; \boldsymbol{\mu}_\star\,,\, \boldsymbol{\Sigma}_\star\right) \end{aligned}$$

then

$$p(y_\star|\mathbf{x}_\star, X, \mathbf{y}) = \iint p(y_\star|f_\star, u_\star)\mathcal{N}\left(\left[\begin{array}{c} u_\star \\ f_\star \end{array}\right]; \boldsymbol{\mu}_\star, \boldsymbol{\Sigma}_\star\right) \mathrm{d}u_\star \mathrm{d}f_\star = Z_R^\star + Z_O^\star.$$

The moments of the posterior are straightforward to calculate using (C.1) and (C.3).

C.3 Ordered overrelaxation for Bernoulli variables

Let the two states be R and O, and let π be the probability of changing state; the marginal probability of the current state is therefore $1-\pi$. With respect to the transition matrix

$$\mathbf{T} = \left[\begin{array}{cc} 1-p & p \\ q & 1-q \end{array}\right],$$

the following relationship must hold:

$$1-\pi = (1-\pi)(1-p) + \pi q \quad \Longrightarrow \quad p = \frac{\pi q}{1-\pi}.$$

The standard solution corresponds to $p = \pi$ and $q = 1-\pi$, for which the successor state is independent of the current state. However, there is no requirement that $p + q = 1$, and ordered overrelaxation decouples the transition probabilities $R \to O$ and $O \to R$ to encourage more frequent changes—clearly maximized when p and q are as large as possible. For this case, the probability of switching states is $\hat{\pi} = \frac{\pi}{1-\pi}$ if $\pi \leq \frac{1}{2}$, and $\hat{\pi} = 1$ if $\pi > \frac{1}{2}$; succinctly,

$$\hat{\pi} = \mathsf{min}\left(1, \frac{\pi}{1-\pi}\right).$$

APPENDIX D

Creating kernel functions

KERNEL FUNCTIONS LIE at the heart of a variety of learning algorithms and probabilistic models: in the learning theory literature they appear in the support vector machine; in a Bayesian setting they feature in the Gaussian process. Their role is to give a numeric value to the similarity between (or correlation in the latent function value at) any two inputs, with the restriction that such valuations respect a certain consistency property that ensures all derived kernel matrices are positive semi-definite. In the absence of any substantial prior knowledge, the relatively benign squared exponential is common; covariances from the Matérn class can be thought of as its rougher cousins; further examples were presented in section 1.2.1.

In this chapter, we explore the merits of a scheme for generating kernels from a class of simple discriminative functions. After introducing the fundamental idea in section D.1 we derive the kernel in section D.2, a discussion of which follows in section D.3. The original motivation for our ideas arose from a consideration of boosting (Schapire, 1990; Freund and Schapire, 1996; Meir and Rätsch, 2003), and we developed an empirically sparse algorithm for binary classification first presented in Naish-Guzman et al. (2005). However, in the following exposition we maintain a purely Bayesian perspective in keeping with the rest of the thesis.

D.1 Kernels from basis functions

Given a class of binary hypotheses $\mathcal{H}$ and an associated prior $p(h)$, we suggest a kernel with the following structure:

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &\doteq \int \mathbb{I}\left[h(\mathbf{x}) = h(\mathbf{z})\right] p(h) \mathrm{d}h - \int \mathbb{I}\left[h(\mathbf{x}) \neq h(\mathbf{z})\right] p(h) \mathrm{d}h \\ &= 1 - 2 \int \mathbb{I}\left[h(\mathbf{x}) \neq h(\mathbf{z})\right] p(h) \mathrm{d}h \in [-1, 1]. \end{aligned} \tag{D.1}$$

In other words, the metric of similarity is the volume of prior probability on $\mathcal{H}$ in which functions disagree at $\mathbf{x}$ and $\mathbf{z}$ subtracted from the volume at which they agree. It is straightforward to prove this is a valid kernel: we need only show that the positive semi-definite property holds for any finite matrix $\mathbf{K} = \{k(\mathbf{x}_n, \mathbf{x}_{n'})\}_{n,n'=1}^{N}$ of dimensions $N \times N$. Observe that, since

$$k(\mathbf{x}, \mathbf{z}) = \int h(\mathbf{x}) h(\mathbf{z}) p(h) \mathrm{d}h, \text{ then for any } \mathbf{v} \in \mathbb{R}^N, \tag{D.2}$$

$$\begin{aligned} \mathbf{v}^T \mathbf{K} \mathbf{v} = \sum_{n=1}^{N} \sum_{n'=1}^{N} v_n v_{n'} k(\mathbf{x}_n, \mathbf{x}_{n'}) &= \int \sum_{n=1}^{N} \sum_{n'=1}^{N} v_n v_{n'} h(\mathbf{x}_n) h(\mathbf{x}_{n'}) p(h) \mathrm{d}h \\ &= \int \left(\sum_{n=1}^{N} v_n h(\mathbf{x}_n) \right)^2 p(h) \mathrm{d}h \geq 0. \end{aligned}$$

This is not a new definition: the form (D.2) appears in Neal (1996) and is used directly in the neural network kernels derived by Williams (1998). The basis functions there take (bounded) real values, which in certain cases yields a tractable integral; Williams treats erf and Gaussian forms. Viewed in relation to that work, we have considered an alternative parameterization and restricted ourselves to binary-valued threshold functions. To obtain the kernel, we need to specify the parameters of elements of $\mathcal{H}$ and place a suitable prior on them, marginalizing for $k(\mathbf{x}, \mathbf{z})$.

D.2 General half-spaces

Initially we restrict our attention to the two-dimensional case, fixing $\mathcal{H}$ to be half-spaces in $\mathbb{R}^2$. Let the origin be $O = (0, 0)$, and let all data lie in the region $[-R, R]^2$. With the exception of those that pass through the origin, linear half-spaces may be paramaterised by a coordinate $\mathbf{r} \in \mathbb{R}^2$ that indicates the closest point on the boundary to the origin O. Let us define the measure $p(h)$ on $\mathcal{H}$ by placing a uniform distribution

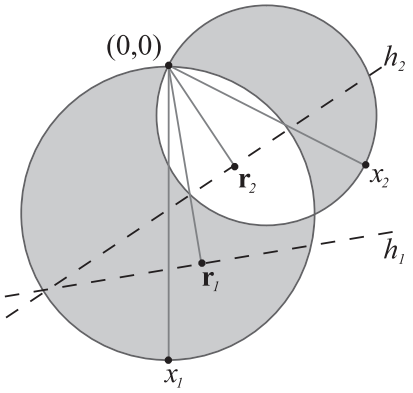


Figure D.1: The shaded region parameterises hypotheses $h \in \mathcal{H}' \subseteq \mathcal{H}$ for which $h(x_1) \neq h(\mathbf{x}_2) \Leftrightarrow h \in \mathcal{H}'$. Two hypotheses are shown, h_1 and h_2, parameterised by $\mathbf{r}_1$ and $\mathbf{r}_2$ respectively. Observe that $h_1 \in \mathcal{H}'$ discriminates between $\mathbf{x}_1$ and $\mathbf{x}_2$, while $h_2 \notin \mathcal{H}'$ classifies the two examples identically.

over $\mathbf{r}$ in the range $[-R, R]^2$. In order to calculate (D.1), we must calculate the volume of parameter space $\mathcal{H}' \subseteq \mathcal{H}$, in which $h(\mathbf{x}) \neq h(\mathbf{z}) \Leftrightarrow h \in \mathcal{H}'$. The situation is illustrated in fig. D.1. We write the circular region parameterising hypotheses that discriminate between a point $\mathbf{x}$ and the origin O as $\bigcirc_{\mathbf{x}}$. Now

$$\begin{aligned}\int \mathbb{I}[h(\mathbf{x}) \neq h(\mathbf{z})]p(h)\mathrm{d}h &\propto |\bigcirc_{\mathbf{x}} \setminus \bigcirc_{\mathbf{z}}| + |\bigcirc_{\mathbf{z}} \setminus \bigcirc_{\mathbf{x}}| \\ &= |\bigcirc_{\mathbf{x}}| + |\bigcirc_{\mathbf{z}}| - 2\,|\bigcirc_{\mathbf{x}} \cap \bigcirc_{\mathbf{z}}|\,. \end{aligned} \tag{D.3}$$

It can be shown that the area of intersection is given by

$$|\bigcirc_{\mathbf{x}} \cap \bigcirc_{\mathbf{z}}| = \frac{1}{2}\Big(\|\mathbf{x}\|^2(\theta_{\mathbf{x}} - \sin\theta_{\mathbf{x}}) + \|\mathbf{z}\|^2(\theta_{\mathbf{z}} - \sin\theta_{\mathbf{z}})\Big),$$

where $\theta_{\mathbf{x}}$ is the angle subtended at the centre of $\bigcirc_{\mathbf{x}}$ by radii extending to the two points of intersection. Using (D.3), we find

$$k(\mathbf{x}, \mathbf{z}; R) = 1 - \frac{2}{R^2}\left(\|\mathbf{x}\|^2(\pi - \theta_{\mathbf{x}} + \sin\theta_{\mathbf{x}}) + \|\mathbf{z}\|^2(\pi - \theta_{\mathbf{z}} + \sin\theta_{\mathbf{z}})\right).$$

There appears here a rather inconvenient "range" parameter R in which the data must lie. We might hope to marginalize a Gaussian prior on elements of $\mathbf{r}$, but due to the complexity of the region over which we must integrate, it is difficult to do this while remaining in a tractable model.

D.2.1 Generalization to higher dimensions

In D dimensions, we can make equivalent prior assumptions and retain tractability. The shapes of interest are D-dimensional hyperspheres, and critical to our kernel is the volume of intersection of any two. Consider two such spheres: we write $\mathbf{c}_1$ and $\mathbf{c}_2$ for the coordinates of their centres; let r_1 and r_2 be their radii. Without loss of generality, let $\mathbf{c}_1 = \mathbf{0}$, and let $r_1 \geq r_2$ and assume there is indeed an intersection. Any point $\mathbf{x} = (x_1, x_2, \ldots, x_D)$ on the surface of sphere i satisfies

$$\sum_{d=1}^{D} (x_d - c_d)^2 = r_i^2.$$

The intersection of the spheres' surfaces is where

$$\|\mathbf{x}\|^2 = r_1^2 \tag{D.4}$$

$$\text{and } \|\mathbf{x} - \mathbf{c}_2\|^2 = r_2^2. \tag{D.5}$$

Again without loss of generality, let the line through the centres of the spheres be the X_1-axis of the Euclidean space. Substituting (D.4) into (D.5) we find

$$r_1^2 - 2\mathbf{x} \cdot \mathbf{c}_2 + \mathbf{c}_2 \cdot \mathbf{c}_2 = r_2^2. \tag{D.6}$$

Now considering only the X_1 axis, the point of intersection is

$$\hat{x} = \frac{r_1^2 - r_2^2 + c_2^2}{2c_2},$$

where we write c_2 as a notational convenience for the X_1 component of $\mathbf{c}_2$. For points x in the range $c_2 - r_2 \leq x < \hat{x}$ on this axis, the volume of intersection is swept out by a $(D-1)$-sphere of radius $s_1(x)$ centred at x. Fix x, and observe that a point $\mathbf{p} = (x, p_2, p_3, \ldots, p_D)$ on the surface of this sub-sphere satisfies

$$\sum_{d=2}^{D} p_d^2 = s_1^2, \qquad \text{and} \qquad (x - c_2)^2 + \sum_{d=2}^{D} p_d^2 = r_2^2,$$

so that the radius is given by $s_1(x)$. Similarly, for coordinates $\hat{x} \leq x \leq r_1$ the volume of intersection is swept out by a $(D-1)$-sphere of radius $s_2(x)$ centred at x, where

$$s_1(x) = \sqrt{r_2^2 - (x - c_2)^2}; \qquad s_2(x) = \sqrt{r_1^2 - x^2}. \tag{D.7}$$

Write the volume of a D-sphere of radius r as $V(D, r)$. Using (D.7), the volume of intersection is

$$V(\mathsf{c}_1 \cap \mathsf{c}_2) = \int_{c_2 - r_2}^{\hat{x}} V(D-1, s_1(x))\mathrm{d}x + \int_{\hat{x}}^{r_1} V(D-1, s_2(x))\mathrm{d}x.$$

In fact, $V(D, r) = \frac{1}{D} S_D r^D$, where $S_D = \frac{2\pi^{D/2}}{\Gamma(D/2)}$, so that

$$\begin{aligned} V(\mathsf{c}_1 \cap \mathsf{c}_2) &= \frac{S_{D-1}}{D-1}\left\{\int_{c_2-r_2}^{\hat{x}} \left(r_2^2 - (x-c_2)^2\right)^{\frac{D-1}{2}}\mathrm{d}x + \int_{\hat{x}}^{r_1}\left(r_1^2 - x^2\right)^{\frac{D-1}{2}}\mathrm{d}x\right\} \\ &= \frac{S_{D-1}}{D-1}\left\{\int_{c_2-\hat{x}}^{r_2} \left(r_2^2 - x^2\right)^{\frac{D-1}{2}}\mathrm{d}x + \int_{\hat{x}}^{r_1}\left(r_1^2 - x^2\right)^{\frac{D-1}{2}}\mathrm{d}x\right\}. \end{aligned}$$

The indefinite integral can be written in analytic form:

$$\int (a - x^2)^{\frac{D-1}{2}} dx = x a^{\frac{D-1}{2}}\, \mathsf{F}\left(\left[\frac{1}{2}, \frac{1-D}{2}\right], \frac{3}{2}, \frac{x^2}{a}\right),$$

where F denotes Gauss' hypergeometric function (Abramowitz and Stegun, 1964, ch. 15).

D.2.2 Implementation

We need to check whether the spheres are entirely disjoint, and also whether the larger sphere entirely encloses the smaller. In either case the limits are not properly defined, although the solutions are trivial (for the former, $V(\mathsf{c}_1 \cap \mathsf{c}_2) = 0$; for the latter, $V(\mathsf{c}_1 \cap \mathsf{c}_2) = V(D, r_2)$). Let $\tilde{\mathbf{c}}_1$ and $\tilde{\mathbf{c}}_2$ define the diameters of two spheres in D dimensions, from a common origin $\tilde{\mathbf{0}}$. The spheres intersect when $\tilde{\mathbf{c}}_1$ and $\tilde{\mathbf{c}}_2$ are not antiparallel; conversely, one entirely encloses the other if they are parallel. Define

$$\mathbf{c}_1 = \mathbf{0} \qquad \text{and} \qquad \mathbf{c}_2 = \frac{\tilde{\mathbf{c}}_2 - \tilde{\mathbf{c}}_1}{2}.$$

Let the radii of the circles be r_1 and r_2, $r_1 \geq r_2$. Adopting the earlier terminology, define the X_1 axis as the line connecting the two centres $\mathbf{0}$ and $\mathbf{c}_2$. By considering (D.6), we see the intersection on this axis is at $\alpha\hat{\mathbf{c}}_2$, where

$$r_1^2 - r_2^2 = 2\alpha(\hat{\mathbf{c}}_2 \cdot \mathbf{c}_2) + \|\mathbf{c}_2\|^2, \tag{D.8}$$

and $\hat{\mathbf{c}}_2$ is a unit vector along $\mathbf{c}_2$. This gives

$$\alpha = \frac{r_1^2 - r_2^2 + \|\mathbf{c}_2\|^2}{2\|\mathbf{c}_2\|}. \tag{D.9}$$

The calculation of the volume of intersection proceeds with $\hat{x} = \alpha$ and $c_2 = \|\mathbf{c}_2\|$.

D.3 Discussion

Our kernel function is closely related to the neural network kernel presented in Williams (1998). In that work, evaluation of the error function

$$h(\mathbf{x}; \mathbf{u}) = \mathrm{erf}\left(u_0 + \sum_{d=1}^{D} u_d x_d\right)$$

is integrated over a prior in which its bias and scale terms are drawn from prespecified Gaussian distributions. Using a clever trick involving the derivative of one of the error functions, Williams is able to obtain a closed form for the correlation between any two points relative to the specified function class, which amounts to the covariance function for an infinite neural network.

In our model, we use the simpler class of threshold functions, parameterized by a coordinate in the space of the data defining the closest point of the linear halfspace to the origin. This is a different prior—although not obviously a better one, certainly in higher dimensions—and yields different correlations. A particular property is non-stationarity, which is also observed in the erf kernel. However, the parameterization does not yield a form which allows a Gaussian expectation, and we are forced to include a range parameter R which defines a region from which we take the infinite limit of uniform samples, and in which the data (including test points) must lie. This is a distinct inconvenience. Furthermore, the generalization to multiple dimensions requires the machinery of Gauss' hypergeometric function—unlike the $\arcsin$ used in the neural network function, this is unlikely to be optimized in floating point units!

Although we obtained in Naish-Guzman et al. (2005) strong results when using our kernel in a form of 1-norm support vector machine, and our classifiers were observed empirically to be very sparse, we must acknowledge certain shortcomings and regard the kernel more as a curiosity than one of enormous practical benefit.

Bibliography

Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions*. Dover, 1964.

Stephen L. Adler. Over-relaxation method for the Monte-Carlo evaluation of the partition function for multiquadratic actions. *Physical Review D – Particles and Fields*, 23 (12):2901–2904, 1981.

Shun-ichi Amari. *Differential-geometrical methods in statistics*, volume 28 of *Lecture Notes in Statistics*. Springer-Verlag, Berlin, 1985.

David F. Andrews and Colin L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society, Series B*, 36(1):99–102, 1974.

Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

Matthew Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

Ted Bergstrom and Mark Bagnoli. Log-concave probability and its applications. *Economic Theory*, 26:445–469, 2005.

Christopher M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

Vladimir I. Bogachev. *Gaussian measures*. American Mathematical Society, 1998.

George E. P. Box and George C. Tiao. A Bayesian approach to some outlier problems. *Biometrika*, 55(1):119–129, 1968.

George E. P. Box and George C. Tiao. *Bayesian inference in statistical analysis*. Addison-Wesley, 1973.

Wray L. Buntine and Andreas S. Weigend. Bayesian back-propagation. *Complex Systems*, 5:603–643, 1991.

Christopher J. C. Burges and Bernhard Schölkopf. Improving the accuracy and speed of support vector machines. In *Advances in Neural Information Processing Systems 9*, pages 375–381. MIT Press, 1997.

Gavin C. Cawley, Nicola L. C. Talbot, Robert J. Foxall, Stephen R. Dorling, and Danilo P. Mandic. Approximately unbiased estimation of conditional variance in heteroscedastic kernel ridge regression. In *Proceedings of the European Symposium on Artificial Neural Networks*, 2003.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20 (3):273–297, 1995.

Richard T. Cox. Probability, frequency, and reasonable expectation. *American Journal of Physics*, 14:1–13, 1946.

Lehel Csató and Manfred Opper. Sparse on-line Gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

Richard P. Feynman. *Statistical Mechanics*. Benjamin, 2 edition, 1972.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

Jerome H. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19 (1):1–67, 1991.

Alan Genz. Numerical computation of rectangular bivariate and trivariate normal and t probabilities. *Statistics and Computing*, 14:151–160, 2004.

Mark N. Gibbs and David J. C. MacKay. Variational Gaussian process classifiers. *IEEE Transactions on Neural Networks*, 11(6):1458–1464, November 2000.

Paul W. Goldberg, Christopher K. I. Williams, and Christopher M. Bishop. Regression with input-dependent noise: a Gaussian process treatment. In *Advances in Neural Information Processing Systems 10*, 1998.

W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6), 1952.

Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pages 5–13. ACM Press, 1993.

Peter J. Huber. *Robust Statistics*. John Wiley and Sons, 1981.

Tommi S. Jaakkola and Michael I. Jordan. Computing upper and lower bounds on likelihoods in intractable networks. In *Proceedings of the Twelfth Conference on Uncertainty in AI*. Morgan Kaufmann, 1996.

Robert A. Jacobs, Michael I. Jordan, Sam J. Nolan, and Geoffrey E. Hinton. Adaptive mixture of local experts. *Neural Computation*, 3:79–87, 1991.

Edwin T. Jaynes. *Probability Theory. The Logic of Science*. Cambridge University Press, 2003.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2): 183–233, 1999.

Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, June 1995.

Kristian Kersting, Christian Plagemann, Patrick Pfaff, and Wolfram Burgard. Most likely heteroscedastic Gaussian process regression. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

Hermann König. *Eigenvalue distribution of compact operators*. Birkhäuser, 1986.

Malte Kuss. *Gaussian process models for robust regression, classification and reinforcement learning*. PhD thesis, Technische Universität Darmstadt, 2006.

Malte Kuss and Carl E. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.

Neil D. Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse Gaussian process methods: the informative vector machine. In *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

Quoc V. Le, Alex J. Smola, and Stéphane Canu. Heteroscedastic Gaussian process regression. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

David J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

David J. C. MacKay. Bayesian model comparison and backprop nets. In *Advances in Neural Information Processing Systems 4*, pages 839–846, 1992a.

David J. C. MacKay. Information based objective functions for active data selection. *Neural Computation*, 4(4):589–603, 1992b.

David J. C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11(5):1035–1068, 1999.

David J. C. MacKay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, 1991.

Ron Meir and Gunnar Rätsch. An introduction to boosting and leveraging. In *Advanced lectures on machine learning*, pages 118–183. Springer-Verlag, 2003.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Thomas Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

Andrew G. P. Naish-Guzman and Sean Holden. The generalized FITC approximation. In *Advances in Neural Information Processing Systems 20*, 2008a.

Andrew G. P. Naish-Guzman and Sean Holden. Robust regression with twinned Gaussian processes. In *Advances in Neural Information Processing Systems 20*, 2008b.

Andrew G. P. Naish-Guzman, Sean Holden, and Ulrich Paquet. On the use of weighted examples in classification. In *Proceedings of the International Conference on Artificial Neural Networks*, 2005.

Subhash C. Narula and John F. Wellington. The minimum sum of absolute errors regression: A state of the art survey. *International Statistical Review*, 50(3):317–326, 1982.

Radford M. Neal. Probabilistic inference using Monte Carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993.

Radford M. Neal. Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation. Technical Report 9508, Dept. of Statistics, University of Toronto, 1995.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Number 118 in Lecture Notes in Statistics. Springer, New York, 1996.

Radford M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report 9702, Dept. of Statistics, University of Toronto, 1997.

Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980.

Anthony O'Hagan and Jon Forster. *Kendall's Advanced Theory of Statistics volume 2B*. Hodder Arnold, 2 edition, 2004.

Manfred Opper and Ole Winther. Gaussian processes for classification: mean field methods. *Neural Computation*, 12(11):2655–2684, 2000.

Manfred Opper and Ole Winther. Expectation consistent approximate inference. *Journal of Machine Learning Research*, 6:2177–2204, 2005.

Ulrich Paquet. *Bayesian inference for latent variable models*. PhD thesis, Computer Laboratory, University of Cambridge, 2007.

Ulrich Paquet, Sean Holden, and Andrew G. P. Naish-Guzman. Bayesian hierarchical ordinal regression. In *Proceedings of the International Conference on Artificial Neural Networks*, 2005.

John C. Platt. Probabilities for SV machines. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

Joaquin Quiñonero-Candela. *Learning with uncertainty—Gaussian processes and relevance vector machines*. PhD thesis, Technical University of Denmark, 2004.

Joaquin Quiñonero-Candela and Carl E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(12): 1939–1959, 2005.

Joaquin Quiñonero-Candela and Carl E. Rasmussen. Analysis of some methods for reduced rank Gaussian process regression. In *Switching and Learning in Feedback Systems*, pages 98–127, 2003.

Joaquin Quiñonero-Candela, Edward Snelson, and Oliver Williams. Sensible priors for sparse Bayesian learning. Technical Report MSR-TR-2007-121, Microsoft Resarch Cambridge, 2007.

Carl E. Rasmussen and Zoubin Ghahramani. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems*, 2002.

Carl E. Rasmussen and Joaquin Quiñonero-Candela. Healing the relevance vector machine through augmentation. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 689–696, New York, 2005. ACM Press.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

Brian Ripley. *Pattern recognition and neural networks*. Cambridge University Press, 1996.

Peter J. Rousseeuw and Annick M. Leroy. *Robust regression and outlier detection*. John Wiley and Sons, 1986.

Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

Matthias Seeger. Expectation propagation for exponential families, 2005. Available from `http://www.cs.berkeley.edu/~mseeger/papers/epexpfam.ps.gz`.

Matthias Seeger. *Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations*. PhD thesis, University of Edinburgh, 2003.

Matthias Seeger, Neil D. Lawrence, and Ralf Herbrich. Sparse Bayesian learning: the informative vector machine. Technical report, Department of Computer Science, Sheffield University, 2002. Available from `http://www.dcs.shef.ac.uk/~neil/papers/`.

Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*. Society for Artificial Intelligence and Statistics, 2003.

Matthias Seeger, Neil D. Lawrence, and Ralf Herbrich. Efficient nonparametric Bayesian modelling with sparse Gaussian process approximations. Available from `http://www.kyb.tuebingen.mpg.de/bs/people/seeger/papers/ivm-jmlr.pdf`, 2006.

Jonathan Richard Shewchuk. Conjugate gradients without the agonizing pain, 1994. Available from `http://www.cs.cmu.edu/~jrs/jrspapers.html`.

Bernard W. Silverman. Some aspects of the spline smoothing approach to nonparametric regression curve fitting. *Journal of the Royal Statistical Society B*, 47: 1–52, 1985.

Alex J. Smola and Peter Bartlett. Sparse greedy Gaussian process regression. In *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.

Alex J. Smola and Bernhard Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

Edward Snelson. *Flexible and efficient Gaussian process models for machine learning*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2007.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*. MIT Press, 2006a.

Edward Snelson and Zoubin Ghahramani. Variable noise and dimensionality reduction for sparse Gaussian processes. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Arlington, Virginia, 2006b. AUAI Press.

Edward Snelson and Zoubin Ghahramani. Local and global sparse Gaussian process approximations. In *Proceedings of the Eleventh International Workshop on Artificial Intelligence and Statistics*, 2007.

Edward Snelson, Carl E. Rasmussen, and Zoubin Ghahramani. Warped Gaussian processes. In *Advances in Neural Information Processing Systems 16*, 2004.

Peter Sollich. Bayesian methods for support vector machines: evidence and predictive class probabilities. *Machine Learning*, 46:21–52, 2002.

Michael E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

Michael E. Tipping and Anita C. Faul. Fast marginal likelihood maximisation for sparse Bayesian models. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

Volker Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

Volker Tresp. Mixtures of Gaussian processes. In *Advances in Neural Information Processing Systems 13*, pages 654–660, 2001.

George E. Uhlenbeck and Leonard Ornstein. On the theory of Brownian motion. *Physical Review*, 36:823–41, 1930.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

Francesco Vivarelli and Christopher K. I. Williams. Discovering hidden features with Gaussian processes regression. In *Advances in Neural Information Processing Systems 11*. MIT Press, 1999.

Grace Wahba, Xiwu Lin, Fangyu Gao, Dong Xiang, Ronald Klein, and Barbara Klein. The bias-variance tradeoff and the randomized GACV. In *Advances in Neural Information Processing Systems 11*, pages 620–626, 1999.

Christopher K. I. Williams. Computation with infinite neural networks. *Neural Computation*, 10(5):1203–1216, 1998.

Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, 2001.