Gaussian Processes for Prediction

Technical Report PARG-07-01

Michael Osborne

Robotics Research Group
Department of Engineering Science
University of Oxford

October 4, 2007

Gaussian Processes for Prediction

Summary

We propose a powerful prediction algorithm built upon Gaussian processes (GPs). They are particularly useful for their flexibility, facilitating accurate prediction even in the absence of strong physical models.

GPs further allow us to work within a complete Bayesian probabilistic framework. As such, we show how the hyperparameters of our system can be marginalised by use of Bayesian Monte Carlo, a principled method of approximate integration. We employ the error bars of our GP's predictions as a means to select only the most informative data to store. This allows us to introduce an iterative formulation of the GP to give a dynamic, on-line algorithm. We also show how our error bars can be used to perform active data selection, allowing the GP to select where and when it should next take a measurement.

We demonstrate how our methods can be applied to multi-sensor prediction problems where data may be missing, delayed and/or correlated. In particular, we present a real network of weather sensors as a testbed for our algorithm.

Contents

Acknowledgements

This work was undertaken as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Systems) project and is jointly funded by a BAE Systems and EPSRC (Engineering and Physical Research Council) strategic partnership.

Work on weather sensor networks is part of an on-going collaboration with Alex Rogers of Southampton University. Similarly, preliminary work on pigeon navigation has been undertaken in collaboration with Richard Mann. Enormous thanks are owed to both my collaborators for their assistance.

Otherwise, I would like to offer huge thanks to my supervisor, Steve Roberts, for many engaging conversations and helpful advice. Similarly, I would like to extend thanks to the erstwhile members of the whole Pattern Analysis and Machine Learning Research Group for their interest and support.

Chapter 1

Introduction

Prediction problems are inescapable. Every day, we implicitly make predictions about how the world will behave, given how we have seen it behave in other times and places. In a more specific sense, prediction is used to refer to the forecasting and tracking of time series data. It is principally problems of prediction in this sense of the word that this report will address. We are motivated by multi-sensor networks, in which we receive a number of related data series. In particular, we will address a weather sensor network.

We abstract these tasks of prediction and tracking as being executed by an *agent*. An agent possesses two faculties: the ability to reason and the power to act. That is, the agent is defined firstly by what knowledge it has, on the basis of which it can employ Bayesian probability theory in order to perform inference about any other unknown quantities. It is defined secondly by its goals, according to which decision theory will uniquely determine which actions it takes. We will principally be employing only the first of these two powers; we consider real problems in which uncertainty is unavoidable.

Such uncertainty takes many forms - data from real sensors is characterised by a myriad of problematic features. Firstly, data will frequently be missing. Worse, such sensor failure is often caused by abnormal events that are of particular interest to us. Where we have multiple sensors, their readings will almost invariably be correlated; the information provided by each is dependent on what readings we have already received. Similarly, the measurements of some sensors may be delayed relative to others.

To tackle these challenges, we consider Gaussian processes. These permit a full probabilistic framework within which we can implement a range of highly flexible models. To enable such flexibility, we must incur a cost: we are forced to introduce a range of hyperparameters. We'll demonstrate how we can integrate out our uncertainty over the values of these hyperparameters by using Gaussian processes a second time. Here we will use a principled form of approximate inference known as Bayesian Monte Carlo.

Gaussian processes are traditionally used for regression on fixed data sets, to produce a single fixed set of predictions. However, we require dynamic and on-line algorithms for our streaming time series data. To meet this requirement, we propose an iterative formulation that will allow us to efficiently update our predictions through time. As the Gaussian process gives us complete probability distributions, we can produce both mean predictions and associated measures of uncertainty. We'll show how such a measure can be used to retain only the most informative data, leading to further gains in the efficiency of our algorithm.

Finally, we are also interested in algorithmically resolving a decision problem: which observation should we take next? Taking an observation is often associated with a non-negligible expenditure of energy, time or both - further, if interested in covertness, an observation will often increase a sensor's risk of detection. Hence it is desirable to minimise the number of observations taken. To solve this problem, we employ the uncertainty of our Gaussian process once again, this time as a measure of the value of an observation. We are then able to take only measurements that are expected to be sufficiently informative.

Chapter 2

Probability Theory

2.1 Foundations

This scenario is fundamentally concerned with how agents should reason under uncertainty. The foundation for research into any such problem is probability theory [Jaynes, 2003]. We explicitly state that we are concerned with a Bayesian approach to probability, in which the probability $P(A|I)$ is interpreted as the degree of belief in proposition A given that proposition I is known to be true. Notationally, I is often used as a 'catch-all' for background information; it represents the sum of all information we possess relevant to the inference at hand. Note we interpret all probabilities as being conditional on the information available in this way.

We also consider any probability to be unique given the conditioning information. That is, $P(A|I)$ unambiguously determines a single probability value, without the need for any subscripts or other definition. Any agent provided with information I, no more and no less, should arrive at the same probability for proposition A. Bayesian probability is subjective only in the sense that results will depend on the information available to the relevant agent.

To further illustrate this point, consider $P(A|I)$ as a measure of the degree to which I logically implies A [Jeffreys, 1998]. That is, this probability is the truth-value associated with the statement $I \Rightarrow A$. If A is logically deducible from I, e.g. if I is the statement $x=1$ while A is the statement $x>0$, the probability is one. Similarly, if the two propositions are inconsistent, e.g. if now A is the statement $x<0$, the resulting probability is zero. Probability values between these two extremes concern the grey area in which it is possible to neither categorically deny nor confirm a proposition with only the information at hand. Probability theory, while being entirely consistent with traditional deductive logic, allows the consideration of a much wider realm of possible propositions. In this sense, probability theory should be viewed as *extended logic*. This extension is far more relevant to the kind of everyday reasoning at which humans are so adept - 'Now, where did I leave my keys?'. For this reason, Bayesian theory is often described as being just common sense, expressed mathematically.

To briefly summarise the mathematical laws of probability[1], for probabilities taken to be real numbers between zero and one:

$$P(\neg A|I) + P(A|I) = 1 \tag{2.1.1a}$$

$$P(A,\, B|I) = P(A|I)\, P(B|A, I) = P(B|I)\, P(A|B, I) \tag{2.1.1b}$$

where $P(A, B|I)$ will be understood to mean the probability of the logical conjunction $P(A \wedge B|I)$. $\neg A$ implies the negation of the proposition A, that is, $\neg A$ is the proposition that A is false. Interestingly, these two operations, conjunction and negation, form a sufficient set of operations to generate all functions of propositions. These laws are important enough to warrant their own names - (2.1.1a) is known as the *Sum Rule* and (2.1.1b) the *Product Rule*.

Bayesian inference is built upon the rearrangement of the Product Rule (2.1.1b)

$$P(B|A, I) = \frac{P(A|B, I)\, P(B|I)}{P(A|I)} \tag{2.1.2}$$

which is known as Bayes' Theorem. $P(B|I)$ is known as the *prior* for B, representing our state of belief about B before learning A. $P(A|B, I)$ is the *likelihood*[2] for B given A, reflecting the impact of the new information A. $P(A|I)$ is the *evidence* for A; a normalisation constant that can be written as $P(A|I) = P(A|B, I)\, P(B|I) + P(A|\neg B, I)\, P(\neg B|I)$ and thus expressed purely in terms of prior and likelihood terms. The combination of these factors thus gives us the *posterior* for B given A, $P(B|A, I)$. This term represents our state of belief about B after having learned A - Bayes' rule allows us to update our probabilities in the light of new information! Bayes' Theorem hence provides a canonical rule for how any agent should reason on the basis of what information it has.

Our analysis will commonly be concerned with the value of some variable. In our notation, uppercase will be used for the propositional variable (presumably unknown), say X, and lowercase for one of its possible values, say x. As en example, X might be 'the mass of the ball' and x might be '0.5 kg'. The set of propositions $\{X = x;\ \forall x\}$, then, are exhaustive and mutually exclusive - that is, one and only one of them is true. We can then consider the probability distribution $P(\,X = \cdot \mid I\,)$ over x. Where clear, X may be dropped to give $P(x|I)$, or $P(\cdot|I)$ if we wish to refer to the distribution itself. We can use the laws of probability (2.1.1) to write the

[1] These are derivable from various reasonable sets of postulates about how degrees of belief should behave, notably those taken by Cox [1946]. Further developments of Cox's ideas have been made by Knuth [2005].

[2] The term likelihood is most often used to describe a function of B for fixed A: $L(B) = P(A|B, I)$. The typical problem of Bayesian inference is to infer some parameter B given experimental data A, hence consideration of a function of the parameter $L(B)$ has significant intuitive appeal. Note, however, that $L(B)$ is not a probability for B; for example, $\int L(B)\, \mathrm{d}B$ will not necessarily be one.

normalisation condition:

$$\sum_x P(\,X\!=\!x\mid I\,) = 1 \tag{2.1.3}$$

and also what is termed the *marginalisation* of a variable Y

$$P(\,X\!=\!x\mid I\,) = \sum_y p(\,X\!=\!x,\,Y\!=\!y\mid I\,) \tag{2.1.4}$$

Similarly, the lower-case $p(\,X\!=\!x\mid I\,)$ will be used to denote the *probability density function* (pdf) for a variable X that may take a continuum of values. Again, X may be dropped for notational convenience. This quantity is defined by

$$p(\,X\!=\!x\mid I\,) \triangleq \lim_{\delta x\to 0}\frac{P(\,x\leq X < x+\delta x\mid I\,)}{\delta x} \tag{2.1.5}$$

As noted by Jaynes [2003], this limit is in general a non-trivial operation. Ignoring it and proceeding as though the laws of probability must necessarily apply to continuous variables can lead to error. However, in practice, so long as we restrict ourselves to finite, normalisable pdfs, we can be justified in using pdfs almost exactly as if they were probabilities [Bretthorst, 1999]. Hence, for infinitesimal $\mathrm{d}x$, we can employ the looser notation:

$$p(\,X\!=\!x\mid I\,)\,\mathrm{d}x \triangleq P(\,x\leq X < x+\mathrm{d}x\mid I\,) \tag{2.1.6}$$

The replacement of the sums in (2.1.3) and (2.1.4) with appropriate integrals give the equivalent relationships for pdfs:

$$1 = \int p(\,X\!=\!x\mid I\,)\;\mathrm{d}x \tag{2.1.7}$$

$$p(\,X\!=\!x\mid I\,) = \int p(\,X\!=\!x,\,Y\!=\!y\mid I\,)\;\mathrm{d}y \tag{2.1.8}$$

However, there are still a few minor hazards to be aware of. Importantly, (2.1.5) clearly implies that $p(\,X\!=\!x\mid I\,)$ is *not* an invariant quantity. As the left hand side of (2.1.5) is a dimensionless probability, and $\mathrm{d}x$ has identical units to x, $p(\,X\!=\!x\mid I\,)$ must have units equal to the inverse of those of x. Hence it is incorrect to apply functions of dimensionless quantities to a pdf of a quantity with units - only if x is dimensionless are expressions such as $\log p(\,X\!=\!x\mid I\,)$ and $\exp p(\,X\!=\!x\mid I\,)$ valid.

We are often interested in making an isomorphic transformation of variables $x\rightarrow y=f(x)$. In this case, our requirement that there be the same probability mass around x and y leads to the expression:

$$\begin{aligned}
P(\,x\leq X < x+\mathrm{d}x\mid I\,) &= P(\,y\leq Y < y+\mathrm{d}y\mid I\,)\\
p(\,X\!=\!x\mid I\,)\;\mathrm{d}x &= p(\,Y\!=\!y\mid I\,)\;\mathrm{d}y\\
p(\,X\!=\!x\mid I\,) &= p(\,Y\!=\!y\mid I\,)\left|\frac{\partial y}{\partial x}\right|
\end{aligned} \tag{2.1.9}$$

Hence in changing variables, the pdf is scaled by the Jacobian of the transformation. An alternative way of seeing the same thing is to use the change of variables $y = f(x')$ in (2.1.8):

$$\begin{aligned} p(\,X\!=\!x \mid I\,) &= \int p(\,X\!=\!x \mid Y\!=\!y,\, I\,)\; p(\,Y\!=\!y \mid I\,)\; \mathrm{d}y \\ &= \int \delta\big(x - f^{-1}(y)\big)\; p(\,Y\!=\!y \mid I\,)\; \mathrm{d}y \\ &= \int \delta\big(x - x'\big)\; p\big(\,Y\!=\!f(x') \mid I\,\big)\; \left|\frac{\partial y}{\partial x'}\right| \mathrm{d}x' \\ &= p(\,Y\!=\!f(x) \mid I\,)\; \left|\frac{\partial y}{\partial x}\right| \end{aligned} \tag{2.1.10}$$

where $\delta(x - a)$ is a Dirac delta density[3] in x centred at a. As an example, if we transform variables as $y = \log x$, a uniform $p(\,Y\!=\!y(x) \mid I\,)$ corresponds to a $p(\,X\!=\!x \mid I\,)$ with form $\frac{1}{x}$.

This sensitivity to representation means that the maximum of a pdf has no status whatsoever in probability theory [MacKay, 2002]. Consider that we have found the extremum x^* of the pdf of a variable X: $\frac{\mathrm{d}}{\mathrm{d}x}\, p(\,X\!=\!x \mid I\,)\big|_{x=x^*} = 0$. Then, if we make a change of variables to $y = f(x)$

$$\begin{aligned} p(\,Y\!=\!y \mid I\,) &= p(\,X\!=\!x \mid I\,)\,\frac{\mathrm{d}x}{\mathrm{d}y} \\ \frac{\mathrm{d}}{\mathrm{d}y}\,p(\,Y\!=\!y \mid I\,) &= p(\,X\!=\!x \mid I\,)\,\frac{\mathrm{d}^2x}{\mathrm{d}y^2} + \frac{\mathrm{d}}{\mathrm{d}x}p(\,X\!=\!x \mid I\,)\left(\frac{\mathrm{d}x}{\mathrm{d}y}\right)^2 \\ \frac{\mathrm{d}}{\mathrm{d}y}\,p(\,Y\!=\!y \mid I\,) &= p(\,X\!=\!x \mid I\,)\,\frac{1}{f''(x)} + \frac{\mathrm{d}}{\mathrm{d}x}p(\,X\!=\!x \mid I\,)\left(\frac{1}{f'(x)}\right)^2 \\ \frac{\mathrm{d}}{\mathrm{d}y}\,p(\,Y\!=\!y \mid I\,)\bigg|_{y=f(x^*)} &= p(\,X\!=\!x^* \mid I\,)\,\frac{1}{f''(x^*)} \neq 0 \end{aligned} \tag{2.1.11}$$

hence the equivalent point $y = f(x^*)$ will, in general, not be an extremum of $p(\,Y\!=\!y \mid I\,)$. Given that there is no reason to discriminate between the representations x and y, we can only conclude that maximising pdfs contains an inevitable element of danger.

2.2 Second-order probability

Consider the archetypical Bernoulli urn. Define X to be the colour of the next ball drawn from the urn. Now imagine two different states of knowledge; I_a, representing the knowledge that ‘the urn contains two million balls, which may be either red or white’ and I_b ‘the urn contains exactly one million white balls and one million red balls’. In both cases, we are equally ignorant about both the possible results of the draw. Neither state of knowledge would lead us to discriminate between $X = \text{Red}$ and $X = \text{White}$. The principle of insufficient

[3] Note that the identical notation $\delta(x - a)$ will also be used to describe a Kronecker delta function in the discrete variable x. Which delta is intended can be simply determined by considering whether the relevant variable is discrete or continuous.

reason applies equally to both, giving

$$P(\,X\!=\!\text{Red}\mid I_a\,) = \frac{1}{2} \tag{2.2.1a}$$

$$P(\,X\!=\!\text{Red}\mid I_b\,) = \frac{1}{2} \tag{2.2.1b}$$

However, I_b is well removed from what we would normally consider ignorance - indeed, it actually represents a great deal of highly pertinent information. It seems reasonable [Jaynes, 2003, Chapter 18] that there is something that distinguishes I_b from I_a. In particular, we feel much more 'confident' in the statement (2.2.1b) than in (2.2.1a). To characterise this 'confidence', consider learning the new piece of information J_d = '10 red balls have just been drawn'. If we had previously known only I_a, it seems reasonable that J_d would lead to suspicions that the urn might contain *only* red balls, or at least a much higher proportion of them. However, given I_b, our probability will be almost entirely unchanged, other than of course being revised slightly downwards to account for the drawn balls. Hence,

$$P(\,X\!=\!\text{Red}\mid I_a,\,J_d\,) \gg \frac{1}{2} \tag{2.2.2a}$$

$$P(\,X\!=\!\text{Red}\mid I_b,\,J_d\,) \simeq \frac{1}{2} \tag{2.2.2b}$$

Hence the assignment (2.2.1b) given I_b seems much more resilient to new information that (2.2.1a) given I_a. So our 'confidence' in a probability seems really dependent on how it would change if given some ancillary information J. We investigate this possibility by defining, for a now arbitrary variable X and arbitrary pieces of information I and J

$$Q(x) \triangleq P(\,X\!=\!x\mid I,\,J\,) \tag{2.2.3}$$

Note that we can consider Q a set of exclusive and mutually exhaustive propositions, using exactly the same notational convention we use for any other variable. That is, $Q = q$ is the proposition that $P(\,X\!=\!x\mid I,\,J\,) = q(x)$, that $P(\,X\!=\!\cdot\mid I,\,J\,)$, uniquely determined by I and J, assumes the particular form $q(\cdot)$. The joint distribution is now represented by Figure 2.1a. As Q is a unique probability distribution given I and J, this node is depicted as deterministic upon I and J.

$$P(\,Q\!=\!q\mid I,\,J\,) = \delta(q - P(\,X\!=\!\cdot\mid I,\,J\,)) \tag{2.2.4}$$

Hence the only uncertainty that can exist about Q is due to uncertainty about J (or I). Given I, Q is just a repackaging of any uncertainty in J. The clear links between this kind of probability Q and information J are stressed by de Finetti [1977].

Note also that Q has no direct influence on X. Imagine programming a probability-robot with information I and J; Q then represents the (two) numbers in the robot's positronic brain associated with its beliefs about X. Clearly, if we already know I and J, Q does not represent any novel information to us. Indeed, if all we know is I and J, Q represents exactly our own beliefs about X! Hence learning Q can not be expected to affect our beliefs about X, given I and J.

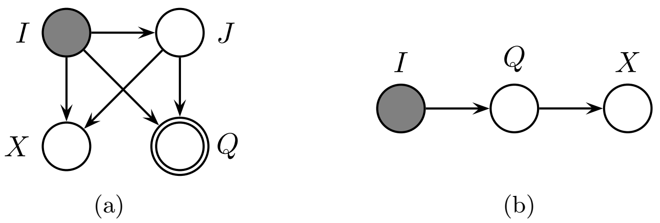


Figure 2.1: Bayesian networks including Q as defined in (2.2.3).

However, now consider that we do not, in fact, know J. Clearly this is the most relevant case to inference - we rarely know exactly what information we might come by in the future. Hence given our lack of certainty about J, the correct thing to do is to marginalise it out.

$$\begin{aligned} p(\, x,\, q \,|\, I\,) &= \int p(\, x \,|\, I,\, j\,)\, p(\, q \,|\, I,\, j\,)\, p(\, j \,|\, I\,)\, \mathrm{d}j \\ &= q(x) \int p(\, q \,|\, I,\, j\,)\, p(\, j \,|\, I\,)\, \mathrm{d}j \\ &= q(x)\, p(\, q \,|\, I\,) \end{aligned} \tag{2.2.5}$$

This new joint is depicted in Figure 2.1b. What have we gained by using this representation? Firstly, note that I affects our belief about X now only through the influence of Q - any part of I or J that is germane to X has been captured by Q. Of course, I and J may still directly influence our beliefs about any variable that is not X. Note that clearly Q is not exerting any causal influence on X - recall that a Bayesian network represents only our epistemic beliefs.

One benefit of this new variable is that we are able to use $p(\, Q\!=\!q \,|\, I\,)$ as a convenient substitute for having to specify a distribution over J. The variable J has been considered a representative of any information we could possibly learn that is pertinent to our beliefs about X. Equivalently, it could also represent any hypothesis or contingency relevant to X. Either way, it is clearly of very high dimension. Our prior over j would likely contain many discontinuities and piecewise components in order to express our beliefs over the enormous space of possible data we could hypothetically receive. It would be very difficult indeed to specify such a distribution. Instead, we have changed variables from J to Q, for which we must specify a distribution over a probability over x - for Boolean X, this requires only a distribution over a single real number between zero and one. The uncertainty in J has been 'folded into' the more convenient Q.

We now illustrate the significance of the prior $p(\, q \mid I\,)$ in the case of the Bernoulli urn example considered earlier. Firstly, note that it is only the mean of this distribution that can in any way affect inference about X

$$\Rightarrow P(\, X = \text{Red} \mid I\,) = \int_0^1 q\, p(\, q \mid I\,)\, \mathrm{d}q \tag{2.2.6}$$

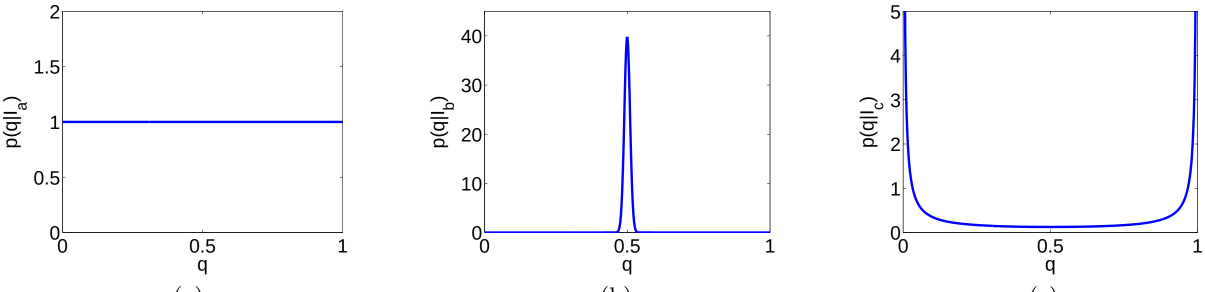
Hence only this first moment of $p(\, q \mid I\,)$ is pertinent to any decisions or inference that in turn depend on X, but not otherwise on any of the factors J. For the urn example, this is the case if we are interested in the colour of a drawn ball, but do not otherwise have any direct interest in factors such as the numbers of each type of ball in the urn, how long and in what manner it was shaken, the local acceleration due to gravity etc. that we can imagine affecting which ball we draw. The higher moments of $p(\, q \mid I\,)$ become relevant only when we consider the impact of learning some of those factors.

Figure 2.2 depicts possible prior distributions for the different states of knowledge we have considered. To extend the example, we have defined an additional possible state of knowledge $I_c =$ 'the urn contains either exclusively white balls or exclusively red balls.' Note firstly that all three distributions have the same mean of one half, as required by (2.2.1). Inasmuch as we are interested only in X, the three possible states of knowledge are indistinguishable. However, as before, consider learning new information. For the diffuse $p(\, q \mid I_a\,)$, we can easily imagine q, our epistemic probability for x, changing to almost any value between zero and one as we receive new data. If we are better informed, as for I_b, we would expect that q would be more resilient to the receipt of new information. Indeed, the peaked shape of $p(\, q \mid I_b\,)$ suggests that it is very improbable that we would learn something that would markedly change our belief q from one half. In the final case of I_c, what we learn will almost certainly push q to close to either one or zero - seeing even a single ball will reveal the entire composition of the urn.

We can be a little more precise about the impact of new data. In particular, consider learning some piece of information K, whilst still being alert to the possibility of learning further information J in the future. Hence, for exactly the reasons presented above, we may wish to proceed by introducing the Q of (2.2.3) to give

$$\begin{aligned} P(\, x \mid I,\, K\,) &= \int q(x)\, p(\, q \mid I,\, K\,)\, \mathrm{d}q \\ &= \frac{\int q(x)\, p(\, q \mid I\,)\; p(\, K \mid q,\, I\,)\, \mathrm{d}q}{\int p(\, q \mid I\,)\; p(\, K \mid q,\, I\,)\, \mathrm{d}q} \end{aligned} \tag{2.2.7}$$

where by use of Bayes' rule we can update our prior distribution $p(\, q \mid I\,)$ to give the required posterior



(a) (b) (c)

Figure 2.2: Second-order probability distributions for the Bernoulli urn problem, given the different states of knowledge:
(a): $I_a =$ 'the urn contains two million balls, which may be either red or white.'
(b): $I_b =$ 'the urn contains exactly one million white balls and one million red balls.'
(c): $I_c =$ 'the urn contains either exclusively white balls or exclusively red balls.'

$p(\, q \,|\, I,\, K\,)$. This can then continue to be updated in this way whenever new information J is obtained.

We now return to the urn and consider repeatedly drawing from it with replacement, as depicted in Figure 2.3. K here represents the observations we make of N drawn balls. In considering repeated trials, we normally wish to assume that our knowledge I is identically pertinent to each trial; it does not discriminate amongst them. Importantly, we also take J to be similarly symmetric in the label of each trial; to represent knowledge that would affect our beliefs about the result of each trial equally. Finally, given both I and J, we also believe the trials to be independent. These conditions are satisfied in our example if J is taken to be the numbers of each type of ball in the urn. We have chosen a set of variables such that we can write

$$P(\,\{x_i;\, i = 1, \ldots, N\} \,|\, I,\, J\,) = \prod_{i=1}^{N} P(\,x_i \,|\, I,\, J\,) \tag{2.2.8}$$

where X_i is the colour of the ith draw and $p(\,X_i = \text{Red} \,|\, I,\, J\,) = q$ is constant in i. Define the proposition that we obtain a particular sequence of exactly n red balls in N draws by

$$\boldsymbol{X}_l = \begin{cases} X_i = \text{Red} & i = l_1, \ldots, l_n \\ X_i = \text{White} & i = l_{n+1}, \ldots, l_N \end{cases} \tag{2.2.9}$$

allowing us to express its probability

$$P(\,\boldsymbol{x}_l \,|\, I,\, J\,) = q^n\,(1-q)^{N-n} \tag{2.2.10}$$

Note that the trials are not independent given I, our initial state of knowledge, alone. $P(\,x_i \,|\, I\,)$ is certainly constant in i; we are equally ignorant about each trial. However, $P(\,x_2 \,|\, I\,) \neq P(\,x_2 \,|\, I,\, x_1\,)$ - as we draw balls, we learn about the composition of the urn, giving us information relevant to the next draw. Including J, assumed constant across trials, has allowed us to express the impact of such observations. This choice of

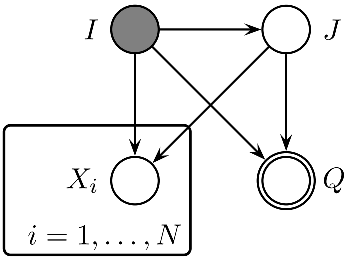


Figure 2.3: Bayesian network for repeated trials X_i.

representation has also given us the use of the convenient Q, with the fortuitous independence structure of (2.2.10).

However, the trials are *exchangeable* given I alone. This means that our joint over trials $P(\, \boldsymbol{X}_l \mid I\,)$ is invariant with respect to permutations of the indices. Effectively, the order of the trials is unimportant; all that is relevant are the number of trials that returned red balls and the number that returned white. That this is true can be seen by writing, using (2.2.10),

$$\begin{aligned} P(\, \boldsymbol{x}_l \mid I\,) &= \iint P(\, \boldsymbol{x}_l \mid I,\, j\,)\, p(\, q \mid I,\, j\,)\, p(\, j \mid I\,)\, \mathrm{d}j\, \mathrm{d}q \\ &= \int q^n \, (1-q)^{N-n}\, p(\, q \mid I\,)\, \mathrm{d}q \qquad (2.2.11) \end{aligned}$$

Interestingly, a theorem of de Finetti [1937] (a proof of which can be found in Heath and Sudderth [1976]) states that for any infinite set of trials for which we possess an exchangeable belief distribution, there exists a variable Q (and associated prior $p(\, q \mid I\,)$) to fill exactly the same role as in (2.2.11). But recall that the existence of Q is nothing more than a rewriting of the presence of J, at least as far as X is concerned. Hence de Finetti's theorem suggests that if our beliefs remain exchangeable even as we imagine taking an infinite number of trials, then there must always exist a set of variables J, which, if learnt, would render all trials independent with identical probability. This theorem underlines the applicability of our methods.

Of course, there are far more factors than simply J, the composition of the urn, that affect the colour of the ball that we draw. Our draw is uniquely determined by a complete specification of how we reach into the urn and the exact position of each ball inside it. However, the experiment has been artificially designed such that these variables are not something we can ever imagine learning about. We are limited to observations of the colours of drawn balls, insufficient information to determine the myriad degrees of freedom of the interior of the urn. When marginalised out, these variables represent no bias towards either red or white - our knowledge of them is insufficient to favour one outcome over the other[4]. Given our unassailable ignorance of all such factors,

[4] Indeed, even given perfect knowledge, the vast *computational* requirements of employing our knowledge would likely preclude accurate prediction.

the best we can hope for is to eliminate in this way the uncertainty due to J, leaving us with a probability of Q.

If we were to somehow learn of some of these variables, our beliefs about the trials would no longer be exchangeable. Knowing the positions of all the balls at a specific point in time represents very pertinent information about the next draw, but much less information about the draw subsequent to that. Hence while we would lose the analytic niceties afforded by exchangeability and Q, our resulting beliefs would be far more precise - giving better predictions as a result.

Now reconsider (2.2.7) in light of Figure 2.3. Define $K = \boldsymbol{X}_l$ and write

$$\begin{aligned} P(\, x \mid I,\, \boldsymbol{x}_l\,) &= \frac{1}{p(\,\boldsymbol{x}_l \mid I\,)} \iint p(\,j \mid I\,)\; P(\,x \mid I,\, j\,)\; P(\,\boldsymbol{x}_l \mid I,\, j\,)\; p(\,q \mid I,\, j\,)\, \mathrm{d}q\, \mathrm{d}j \\ &= \frac{\int q^{n+1}\,(1-q)^{N-n}\, p(\,q \mid I\,)\, \mathrm{d}q}{\int q^{n}\,(1-q)^{N-n}\, p(\,q \mid I\,)\, \mathrm{d}q} \end{aligned} \tag{2.2.12}$$

Note that (2.2.12) would be identical if K had instead been simply an observation of the numbers of red and white balls - binomial coefficients would have cancelled from both numerator and denominator. As an example, we can investigate the diffuse $p(\,q \mid I\,) = 1$ depicted in Figure 2.2a, expressing total ignorance about the proportions of red and white balls. For this case, we employ the complete Beta function to obtain

$$P(\,x \mid I,\, \boldsymbol{x}_l\,) = \frac{n+1}{N+2} \tag{2.2.13}$$

which is Laplace's rule of succession, discussed in depth in Jaynes [2003, Chapter 18].

The concept of 'physical' probabilities or 'propensities' is often met in dealings with repeated trials. The discussion above makes it clear why some may be tempted to adopt such notions - for exchangeable trials, we can indeed find a probability Q that applies equally to each trial. However, we stress again that this probability is purely and simply an expression of our beliefs assuming knowledge of J, the variables that act identically on each trial. Inference performed on Q itself is useful, so long as we are clear that it is no more than a placeholder for J. Our previous depiction of Q as being a physical object in a suitably informed robot's mind is often helpful as an *intuition pump* [Dennett, 1991] for how Q interacts with other variables. Indeed, an important use of Q emerges when we consider multi-agent systems, when our agents will need to possess beliefs about what is going on inside other agents' heads. But these probabilities Q need not be 'physical' for us to possess a belief about them. A probabilistic agent is entitled to a belief about *any* proposition. A *second-order* probability is a probability of probability - a concept that has met with much opposition, as discussed by Goldsmith and Sahlin [1983]. We propose, however, that the concept is widely useful and intuitive - so long as we don't let our intuition confuse as to what we are really doing.

Chapter 3

Gaussian Processes

3.1 Introduction

Gaussian processes [Rasmussen and Williams, 2006, MacKay, 1998] represent a powerful way to perform Bayesian inference about functions. We begin by considering functions as vectors, a long list of all possible function outputs that is associated with a similarly long list of all possible function inputs. For most useful functions, however, the number of possible inputs is infinite - we typically want to consider functions over the natural or real numbers. To manage this potential difficulty, we define a Gaussian process (GP) as being a probability distribution over a (possibly infinite) number of variables, such that the distribution over any finite subset of them is a multi-variate Gaussian. We are now free to use a GP as a prior distribution for a function.

This choice provides a truly remarkable degree of flexibility, as we'll see later. However, our choice is undeniably informed by the remarkably expedient properties of the Gaussian distribution (Appendix A.1). In particular, the marginals of a multivariate Gaussian distribution are themselves Gaussian. Similarly, the distribution of any subset of variables conditioned upon any other subset is also Gaussian. These properties allow us to readily consider subsets of the potentially infinite lists of function outputs and inputs. With these subsets representing finite sets of observed predictors and desired predictants, Gaussianity allows us to simply ignore any values of the function that we are neither interested in nor knowledgeable about.

We consider a function $x(t)$. As we are uncertain about the actual values our function takes, we must treat them as we would any other 'random variable'. We consider propositions such as $X(t) = x(t)$ - the actual function at t assumes the possible value $x(t)$. We are principally concerned with prediction and hence functions in time, but in all that follows t may equally interpreted as any arbitrary kind of input. We'll use $\boldsymbol{x}$ to refer to a possible vector of function outputs and $\boldsymbol{t}$, function inputs. I will be used to represent the models and background knowledge that we employ in our analysis. A Bayesian network depicting the possible correlations amongst our variables is depicted in Figure 3.1. Hence we define a GP as the distribution

$$p(\,\boldsymbol{x} \mid \boldsymbol{t},\, \boldsymbol{\mu},\, K,\, I\,) \triangleq \mathbf{N}(\boldsymbol{x};\, \boldsymbol{\mu},\, K) \tag{3.1.1}$$

which by the properties of a Gaussian will hold true for some mean $\boldsymbol{\mu}$ and covariance K regardless of the subset represented by $\boldsymbol{x}$ and $\boldsymbol{t}$. Clearly it is the choice of the mean and covariance that defines a GP [MacKay, 2006]; the two respectively defining location and scale in a way identical to any more familiar Gaussian distribution.

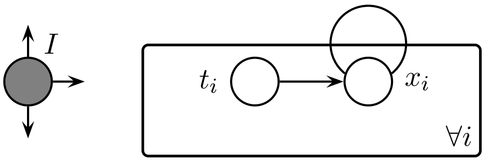


Figure 3.1: Bayesian network for a generic GP - note the value of the function $x(\cdot)$ at a particular input may potentially be related to its value for any other input.

3.2 Parameters and Hyperparameters

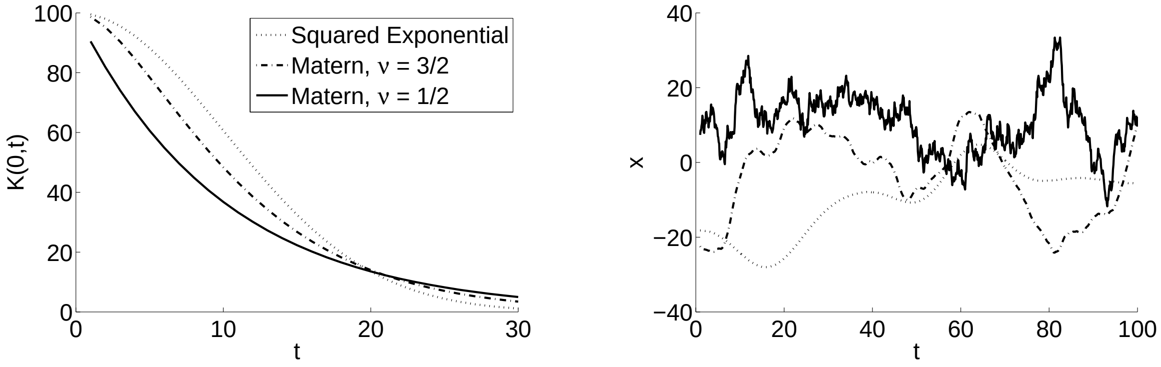
The mean here is the function we expect before making any observations. A popular, simple choice is to take $\boldsymbol{\mu} = \mathbf{0}$, expressing that our initial best guess for the function output at any input is zero. Of course, if we possess better knowledge than this assignment, it should be included. One possibility is to take $\boldsymbol{\mu}$ as a non-zero constant for all inputs, where this constant forms a hyperparameter about which we will also possess prior beliefs. If we are willing to consider more hyperparameters, we could take the mean as any arbitrary parameterised function. Henceforth, we'll refer to the set of hyperparameters of our model as ϕ. Marginalising these hyperparameters, effectively nuisance variables, forms the essence of the challenge of GP prediction.

The covariance must similarly be defined in terms of hyperparameters. K is a square matrix indicating the strength of correlations amongst the entries of $\boldsymbol{x}$. Hence it must have an appropriate entry for every possible pair of admissable inputs, where the number of admissable inputs will typically be infinite. If the mean is a curve, the covariance is a surface; it is generated by a *covariance function*. The difficulty with parameterising such a function is due to the requirement that K be positive semi-definite; regardless, many choices exist [Abrahamsen, 1997, Gibbs, 1997, Gneiting, 2002].

Almost all functions of interest we expect to possess some degree of smoothness. That is, the value of a function at t is strongly correlated with the values close to t, these correlations becoming weaker further away. A further common assumption is that of stationarity, which takes the correlation between points to be purely a function of the difference in their inputs, $t - t'$. This effectively asserts that our function looks more or less similar throughout its domain. Similarly, we will often want our function to be isotropic, such that it does not have a preferred direction. In this case, the covariance will be a function of the separation in input space, $|t - t'|$. There are many covariance functions we can choose to satisfy these requirements, all giving slightly different weightings of correlations with distance. A prototypical choice is the squared exponential

$$K_\phi(t,t') \triangleq h^2 \exp\left(-\frac{1}{2}\left|\frac{t-t'}{w}\right|^2\right) \tag{3.2.1}$$

Here $h > 0$ and $w > 0$, elements of ϕ, specify the expected length scales of the function in output ('height') and input ('width') spaces respectively. These hyperparameters are common to all stationary isotropic covariance functions. Note that as $w \to 0$, the function evaluated at any discrete set of points will tend towards the appearance of pure noise. As $w \to \infty$, the function will appear infinitely smooth, that is, flat. For intermediate w, the squared exponential is notable for giving rise to particularly smooth functions.



(a) (b)

Figure 3.2: (a) The covariance as a function of separation and (b) pseudo-random draws from the relevant GP for squared exponential and Matérn covariance functions. Note the strong correlations associated with the smoother functions. For all cases, $\mu = 0$, $w = 10$ and $h = 10$.

However, it has been argued [Stein, 1999] that the strong smoothness that (3.2.1) enforces is an inappropriate assumption for many physical processes. A more flexible alternative is the Matérn class of covariance functions, given by

$$K_\phi(t,t') \triangleq h^2 \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\left|\frac{t-t'}{w}\right|\right)^\nu \mathfrak{K}_\nu\left(\sqrt{2\nu}\left|\frac{t-t'}{w}\right|\right) \tag{3.2.2}$$

where $\nu > 0$ is a smoothness hyperparameter (larger ν implies smoother functions) and $\mathfrak{K}_\nu$ is the modified Bessel function. Fortunately, (3.2.2) simplifies for half-integer ν, to give, for the example of $\nu = \frac{3}{2}$

$$K_{\phi,a,\nu=3/2}(t,t') = h^2\left(1+\sqrt{3}\left|\frac{t-t'}{w}\right|\right)\exp\left(-\sqrt{3}\left|\frac{t-t'}{w}\right|\right) \tag{3.2.3}$$

These covariance functions are illustrated in Figure 3.2.

3.3 Modifying Covariance Functions

Given such covariance functions as building blocks, there are many ways [MacKay, 1998, Rasmussen and Williams, 2006] to construct valid new covariance functions by modifying and combining them. Such modifications permit us to express more sophisticated beliefs about the structure of the function of interest. For example, if we know that the function under consideration $x(t)$ is actually the sum of independent but unknown functions $a(t)$ and $b(t)$, the probability calculus gives

$$\begin{aligned} p(\,\boldsymbol{x} \,|\, \boldsymbol{t},\, \boldsymbol{\mu}_A,\, \mathbf{K}_A,\, \boldsymbol{\mu}_B,\, \mathbf{K}_B,\, I\,) &= \iint \delta(\boldsymbol{x} - \boldsymbol{a} + \boldsymbol{b})\; p(\,\boldsymbol{a} \,|\, I\,)\; p(\,\boldsymbol{b} \,|\, I\,)\; \mathrm{d}\boldsymbol{a}\, \mathrm{d}\boldsymbol{b} \\ &= \int \mathbf{N}(\boldsymbol{a};\, \boldsymbol{\mu}_A,\, \mathbf{K}_A)\; \mathbf{N}(\boldsymbol{x} - \boldsymbol{a};\, \boldsymbol{\mu}_B,\, \mathbf{K}_B)\, \mathrm{d}\boldsymbol{a} \\ &= \mathbf{N}(\boldsymbol{x};\, \boldsymbol{\mu}_A + \boldsymbol{\mu}_A,\, \mathbf{K}_A + \mathbf{K}_B) \end{aligned} \tag{3.3.1}$$

Hence the GP over this $x(t)$ has a covariance that is simply the sum of the covariances for its two constituents. In a similar vein, consider a function $x(t)$ known to be the product of two independent functions $a(t)$ and $b(t)$. Note that this means that $\boldsymbol{x}$ is the Hadamard, or element by element, product $\boldsymbol{x} = \boldsymbol{a} \bullet \boldsymbol{b}$. We can approximate the distribution of $\boldsymbol{x}$ by taking only its first two moments to give the GP

$$p(\,\boldsymbol{x} \,|\, \boldsymbol{t},\, \boldsymbol{\mu}_A,\, \mathbf{K}_A,\, \boldsymbol{\mu}_B,\, \mathbf{K}_B,\, I\,) \;\simeq\; \mathbf{N}\big(\boldsymbol{x};\, \boldsymbol{\mu}_A \bullet \boldsymbol{\mu}_A,\, \mathbf{K}_A \bullet \mathbf{K}_B + \mathbf{K}_A \bullet \boldsymbol{\mu}_B \boldsymbol{\mu}_B^{\mathrm{T}} + \mathbf{K}_B \bullet \boldsymbol{\mu}_A \boldsymbol{\mu}_A^{\mathrm{T}}\big) \tag{3.3.2}$$

So long as the magnitude of our uncertainties, represented by the covariances, are not too large, the approximation is reasonably good[1]. Hence we can take the dot product of two covariances as a valid covariance in its own right. Similarly, the tensor product, including particularly the Kronecker product, of two covariance functions also forms a valid covariance function in the appropriate product space. Finally, a covariance under any arbitrary map $t \longrightarrow u(t)$ remains a viable covariance. A particularly relevant example of this allows us to modify our stationary covariance functions to model periodic functions. In this case, we can simply replace any factors of $\left|\frac{t-t'}{w}\right|$ by $\sin \pi \left|\frac{t-t'}{w}\right|$.

3.4 Correlated Inputs and Outputs

The stationary, isotropic covariance functions (3.2.1), (3.2.2) and (3.2.3) listed above were stated for the simplest case of one dimensional inputs and outputs. However, it is not difficult to extend covariances to allow for multiple input or output dimensions. Recall that w and h were our scales in input and output spaces - if these are multi-dimensional, one approach is to simply define multi-dimensional analogues of the scales.

[1] The third central moment is proportional to $(\boldsymbol{\mu}_A \otimes \mathbf{K}_A) \bullet (\boldsymbol{\mu}_B \otimes \mathbf{K}_B)$, where $\otimes$ represents the Kronecker product.

For a multi-dimensional input space, this means replacing the previously employed dimensionless notion of separation $\left|\frac{t-t'}{w}\right|$ by the more general $\|t-t'\|_{\mathbf{w}}$. Recall from above that the covariance produced by such a transformation remains valid. Both w and $\mathbf{w}$ form a parameterisation of a relevant dimensionless norm; we now consider $\mathbf{w}$ as a square matrix of suitable dimension. Note that a matrix $\mathbf{B}$ that can be expressed as $\mathbf{B} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$, for any matrix $\mathbf{A}$, is clearly positive semi-definite. Hence an example of a suitable norm is

$$\|t-t'\|_{\mathbf{w}} = \sqrt{(t-t')^{\mathrm{T}}\left(\mathbf{w}^{\mathrm{T}}\mathbf{w}\right)^{-1}(t-t')} \tag{3.4.1}$$

If $\mathbf{w}$ is a diagonal matrix, its role in (3.4.1) is simply to provide an individual scale for each dimension. An illustration of what it means to have individual scales in this way is provided by Figure 3.3. However, by introducing off-diagonal elements, we can allow for correlations amongst the input dimensions.

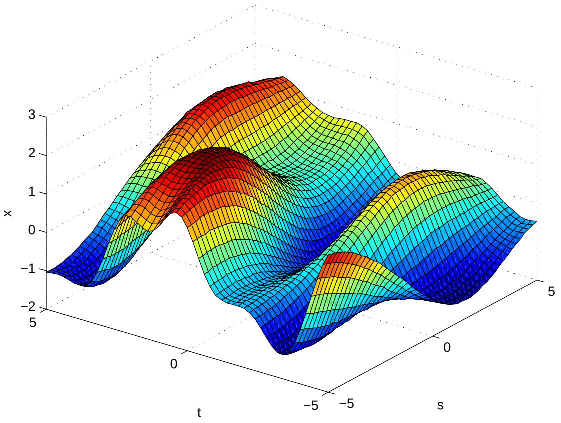


Figure 3.3: Pseudo-random draw from a GP for a function x over two inputs, s and t. $\mathbf{w}$ is taken to be diagonal with a scale of 3 in s and 1 in t.

Similarly, multi-dimensional output spaces can be dealt with by the use of a transformation. In this case, we simply replace h^2 with $\mathbf{h}^{\mathrm{T}}\mathbf{h}$, where $\mathbf{h}$ is a square matrix of appropriate dimension. In an identical fashion to the properties of $\mathbf{w}$, a diagonal $\mathbf{h}$ specifies merely an independent scale for each output. Note that $\mathbf{h}^{\mathrm{T}}\mathbf{h}$ is a valid covariance over the space of output labels. That is, it specifies any direct correlations amongst the output variables. The full covariance is then a Kronecker product of $\mathbf{h}^{\mathrm{T}}\mathbf{h}$ and a covariance function with a single output variable. As stated earlier, this tensor product forms a viable covariance.

One as yet unresolved issue with using these multi-dimensional methods is the parameterisation of $\mathbf{h}$ and $\mathbf{w}$. In particular, while both $\mathbf{h}^{\mathrm{T}}\mathbf{h}$ and $\mathbf{w}^{\mathrm{T}}\mathbf{w}$ are positive semi-definite, as required, the correlations they embody are not intuitively connected with the entries of $\mathbf{h}$ and $\mathbf{w}$. If we use the entries of $\mathbf{h}$ and $\mathbf{w}$ directly as our hyperparameters, it is not clear what priors should be assigned to them in order to represent our expected correlations. Fortunately, other parameterisations exist [Pinheiro and Bates, 1996].

In particular, we employ the *spherical parameterisation*, in which we write either $\mathbf{h}$ or $\mathbf{w}$ in the form $\mathbf{s}\,\mathrm{diag}(\boldsymbol{l})$. Here $\mathbf{s}$ is an upper triangular matrix, whose nth column contains the spherical coordinates in $\mathbf{R}^n$ of a point on the hypersphere $\mathbf{S}^{n-1}$, followed by the requisite number of zeros. As an example, $\mathbf{s}$ for a four dimensional space is

$$\mathbf{s} = \begin{bmatrix} 1 & \cos\theta_1 & \cos\theta_2 & \cos\theta_4 \\ 0 & \sin\theta_1 & \sin\theta_2\,\cos\theta_3 & \sin\theta_4\,\cos\theta_5 \\ 0 & 0 & \sin\theta_2\,\sin\theta_3 & \sin\theta_4\,\sin\theta_5\,\cos\theta_6 \\ 0 & 0 & 0 & \sin\theta_4\,\sin\theta_5\,\sin\theta_6 \end{bmatrix} \tag{3.4.2}$$

This form ensures that $\mathbf{s}^{\mathrm{T}}\mathbf{s}$ has ones across its diagonal and hence all other entries may be thought of as akin to correlation coefficients, lying between -1 and 1. Meanwhile, $\boldsymbol{l}$ is a vector of the length scales of each dimension, defining something like the standard deviations of each variable. In fact, for the parameterisation of $\mathbf{w}$, we can remove the phrases 'akin to' and 'something like' if we consider a squared exponential covariance function with the norm (3.4.1). Similarly, for the parameterisation of $\mathbf{h}$, these are exactly correlation coefficients and standard deviations for the outputs for any single input. If we consider varying the input, the correlations will be simply multiplied by a factor given by the appropriate covariance function. These connections provide the motivation for the use of this parameterisation. Finally, note that the total number of hyperparameters required by the spherical parameterisation is $\frac{1}{2}\,N\,(N+1)$ for an input space of dimension N. Clearly this is the same as parameterising the elements of an upper triangular $\mathbf{h}$ or $\mathbf{w}$ directly. The spherical parametrisation captures all possible covariance matrices, but requires a large number of hyperparameters to give this generality.

Of course, there are other covariance structures that may be more appropriate in light of our knowledge of the system under consideration. In particular, we consider now an alternative to forcing the covariance for multiple outputs to be the Kronecker product of a fixed covariance over output label and a covariance over the inputs. Consider the case where we suspect that the outputs might be all distinctly corrupted versions of an unobserved function $a(t)$ of lower dimension. As a reasonably general model, we write the outputs as $x(t) = b(t) + M\,a(t)$. Here M is a linear mapping that transforms $a(t)$ into the appropriate higher dimensional space of the outputs $x(t)$. $b(t)$ gives the high dimensional corrupting function. By placing independent GPs on $\boldsymbol{b}$, $\mathbf{N}(\boldsymbol{a};\, \boldsymbol{\mu}_A,\, \mathbf{K}_A)$, and $\boldsymbol{a}$, $\mathbf{N}(\boldsymbol{b};\, \boldsymbol{\mu}_B,\, \mathbf{K}_B)$, we can use exactly (A.1.7) to give

$$p(\,\boldsymbol{x} \mid \boldsymbol{t},\, \boldsymbol{\mu}_A,\, \mathbf{K}_A,\, \boldsymbol{\mu}_B,\, \mathbf{K}_B,\, I\,) = \mathbf{N}\big(\boldsymbol{x};\, M\,\boldsymbol{\mu}_A + \boldsymbol{\mu}_B,\, \mathbf{K}_B + M\,\mathbf{K}_A\,M^{\mathrm{T}}\big) \tag{3.4.3}$$

As an example, consider sensors labelled $n = 1, \ldots, N$ in fixed positions. They each make readings $x_n(t)$ of a single underlying variable $a(t)$, say, the light intensity from the sun. However, the sensors' readings are corrupted by a function $b(t)$, such as the cloud cover, which varies both spatially and temporally. M in this case will simply be multiple copies of the identity, $\mathbf{1}_N \otimes \mathbf{I}$. Our GP over $b(t)$ will allow us to explicitly model the correlations amongst the noise contributions for each sensor. For example, it seems reasonable that the cloud cover will be similar for neighbouring sensors. The additive covariance structure of (3.4.3) has allowed an appropriate and flexible model for this multi-dimensional system. Another, less direct, approach to correlated data streams is given by Boyle and Frean [2005].

3.5 Implementation

Given a set of hyperparameters ϕ, then, we can evaluate a covariance $\mathbf{K}_\phi(t, t')$ and mean $\mu_\phi(t)$. Now consider additionally knowing the predictor data $\boldsymbol{x}_D$ at $\boldsymbol{t}_D$ and being interested in the values of the predictants $\boldsymbol{x}_\star$ at known $\boldsymbol{t}_\star$. Using (3.1.1) and the properties of the Gaussian distribution (Appendix A.1), we can write

$$p(\, \boldsymbol{x}_\star \,|\, \boldsymbol{t}_\star,\, \boldsymbol{x}_D,\, \boldsymbol{t}_D,\, \phi,\, I\,) = N\Big(\boldsymbol{x}_\star; \mu_\phi(\boldsymbol{t}_\star) + \mathbf{K}_\phi(\boldsymbol{t}_\star, \boldsymbol{t}_D)\, \mathbf{K}_\phi(\boldsymbol{t}_D, \boldsymbol{t}_D)^{-1}\, (\boldsymbol{x}_D - \mu_\phi(\boldsymbol{t}_D)),$$
$$\mathbf{K}_\phi(\boldsymbol{t}_\star, \boldsymbol{t}_\star) - \mathbf{K}_\phi(\boldsymbol{t}_\star, \boldsymbol{t}_D)\, \mathbf{K}_\phi(\boldsymbol{t}_D, \boldsymbol{t}_D)^{-1}\, \mathbf{K}_\phi(\boldsymbol{t}_D, \boldsymbol{t}_\star)\Big) \quad (3.5.1)$$

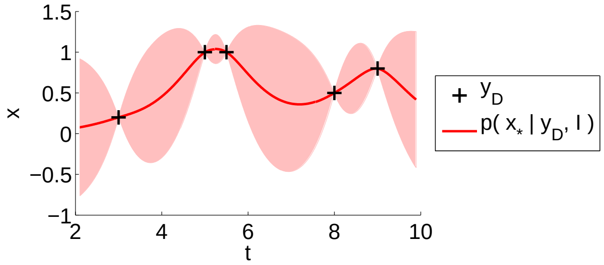


Figure 3.4: A prototypical example of GP inference using a Matérn covariance with $\nu = \frac{3}{2}$, $w = 1$ and $h = 1$. The dark line is the mean of the posterior GP, the shaded area contains plus or minus a single standard deviation as an illustration of its uncertainty.

Equation (3.5.1) forms the cornerstone of GP inference - it permits us to update our beliefs about the predictants $\boldsymbol{x}_\star$, after having obtained data $\boldsymbol{x}_D$. An example of using these equations for inference is demonstrated in Figure 3.4. In using GPs for regression, $\boldsymbol{t}_\star$ will typically lie amongst the set $\boldsymbol{t}_D$. For prediction, we use data from past times $\boldsymbol{t}_D$ to make predictions about future times $\boldsymbol{t}_\star$.

In most real applications, though, we observe not the underlying variable of interest x, but only some noise-corrupted version of it, y. GPs are unsurprisingly readily extended to allow for Gaussian noise models[2], in which we assume a Gaussian likelihood $p(\, \boldsymbol{y} \,|\, \boldsymbol{x}\, I\,)$. In particular, we assume independent Gaussian noise contributions of a fixed variance σ^2. This noise variance effectively becomes another hyperparameter of our

[2] Other noise models are possible, at the cost of the loss of the analytic niceties associated with Gaussianity. The study of other noise models forms an active area of research, see Rasmussen and Williams [2006].

model, and will ultimately be coupled to a prior and marginalised. To proceed, we define

$$\mathbf{V}_\phi(t,t') \triangleq \mathbf{K}_\phi(t,t') + \sigma^2\,\delta\big(t-t'\big) \tag{3.5.2}$$

where $\delta(-)$ is the Kronecker delta function. Henceforth, where clear, the subscript ϕ will be dropped for notational convenience. Further, we will assume that the relevant inputs $\boldsymbol{t}_D$ and $\boldsymbol{t}_\star$ are always known to us and as such will assimilate them into I. Hence, by using the properties of the Gaussian distribution (Appendix A.1) once again, we have

$$\begin{aligned} p(\,\boldsymbol{x}_\star \,|\, \boldsymbol{y}_D,\,\phi,\,I\,) = N\Big(&\boldsymbol{x}_\star; \mu(\boldsymbol{t}_\star) + \mathbf{K}(\boldsymbol{t}_\star,\boldsymbol{t}_D)\,\mathbf{V}(\boldsymbol{t}_D,\boldsymbol{t}_D)^{-1}\,(\boldsymbol{y}_D - \mu(\boldsymbol{t}_D)),\\ &\mathbf{K}(\boldsymbol{t}_\star,\boldsymbol{t}_\star) - \mathbf{K}(\boldsymbol{t}_\star,\boldsymbol{t}_D)\,\mathbf{V}(\boldsymbol{t}_D,\boldsymbol{t}_D)^{-1}\,\mathbf{K}(\boldsymbol{t}_D,\boldsymbol{t}_\star)\Big) \end{aligned} \tag{3.5.3}$$

The implementation of (3.5.3) is particularly sensitive to the condition number of $\mathbf{V}$ - this is the ratio of its largest eigenvalue to its smallest. Unfortunately, a smooth covariance is necessarily poorly conditioned. Smoothness requires that two nearby points are strongly correlated, and so the two rows/columns in the covariance matrix corresponding to those points will both be very similar. If we imagine those two points becoming progressively closer, the matrix will consequently tend towards singularity. We can think of this being the complaint of the GP that it is being given increasingly contradictory information, as if it were being told the function had two distinct values at a single point. This problem becomes particularly acute in the otherwise desirable case that we have a large, finely separated set of almost noiseless data.

One way to counterract this problem is to tell the GP that there is more noise than there actually is, by adding a small positive quantity known as *jitter* [Neal, 1997] to the diagonals of $\mathbf{V}$. Alternatively, we can use a less smooth covariance function - this gives another reason for the use of the flexibly smooth Matérn covariance. We should certainly not make our covariance enforce more smoothness than we actually believe exists! Finally, we could choose to discard data points as necessary to obtain a data set that is well-separated as adjudged by the input scale $\mathbf{w}$. These options will all artifically decrease the condition number and reduce associated errors.

However, easily the most important action to take in minimising the consequences of poor conditioning is to avoid the explicit computation of the matrix inverse. A vastly more stable alternative is the computation of the Cholesky decomposition $\mathbf{R}(\boldsymbol{t}_D,\boldsymbol{t}_D)$ of $\mathbf{V}(\boldsymbol{t}_D,\boldsymbol{t}_D) = \mathbf{R}(\boldsymbol{t}_D,\boldsymbol{t}_D)^{\mathrm{T}}\,\mathbf{R}(\boldsymbol{t}_D,\boldsymbol{t}_D)$. We denote the Cholesky operation as $\mathbf{R}(\boldsymbol{t}_D,\boldsymbol{t}_D) = \mathrm{chol}(\mathbf{V}(\boldsymbol{t}_D,\boldsymbol{t}_D))$. This upper triangular factor can then be used to efficiently solve our required triangular sets of linear equations[3]. For $\mathbf{A}$ triangular, we define $\boldsymbol{x} = \mathbf{A} \setminus \boldsymbol{b}$ as the solution to the

[3] It's worth noting that there are also fast techniques that allow an approximate solution to these equations that do not require the computationally expensive calculation of the Cholesky factor at all. See, for example, Neal [1997]. Given the techniques that will be described later, however, we do not believe such an approximation is necessary.

equations $\mathbf{A}\,\boldsymbol{x} = \boldsymbol{b}$ as found by the use of backwards or forwards substitution. Note that this operation must be performed twice for (3.5.3), to give two terms we define as

$$\mathfrak{a}_D \triangleq \mathbf{R}(\boldsymbol{t}_D, \boldsymbol{t}_D)^{\mathrm{T}} \setminus (\boldsymbol{y}_D - \mu(\boldsymbol{t}_D)) \tag{3.5.4}$$

$$\mathfrak{b}_{\star,D} \triangleq \mathbf{R}(\boldsymbol{t}_D, \boldsymbol{t}_D)^{\mathrm{T}} \setminus \mathbf{K}(\boldsymbol{t}_D, \boldsymbol{t}_\star) \tag{3.5.5}$$

Rewriting (3.5.3)

$$p(\, \boldsymbol{x}_\star \,|\, \boldsymbol{y}_D,\, \phi,\, I\,) = \mathrm{N}\Big(\boldsymbol{x}_\star; \mu(\boldsymbol{t}_\star) + \mathfrak{b}_{\star,D}^{\mathrm{T}}\, \mathfrak{a}_D, \mathbf{K}(\boldsymbol{t}_\star, \boldsymbol{t}_\star) - \mathfrak{b}_{\star,D}^{\mathrm{T}}\, \mathfrak{b}_{\star,D}\Big) \tag{3.5.6}$$

3.6 Marginalising Hyperparameters

We now return to the problem of marginalising our hyperparameters. Firstly, we need to assign appropriate priors $p(\,\phi\,|\,I\,)$. Fortunately [Gibbs, 1997], GP hyperparameters tend to be favoured with intuitive physical significance, thus assisting the elicitation of such priors. Given these, the expression we need to evaluate is:

$$p(\, \boldsymbol{x}_\star \,|\, \boldsymbol{y}_D,\, I\,) = \frac{\int p(\, \boldsymbol{x}_\star \,|\, \boldsymbol{y}_D,\, \phi,\, I\,)\; p(\, \boldsymbol{y}_D \,|\, \phi,\, I\,)\; p(\,\phi\,|\,I\,)\; \mathrm{d}\phi}{\int p(\, \boldsymbol{y}_D \,|\, \phi,\, I\,)\; p(\,\phi\,|\,I\,)\; \mathrm{d}\phi} \tag{3.6.1}$$

(3.6.1) performs a great deal of work for us; it's worth examining its powers in more detail. In particular, we haven't yet addressed a method for choosing the level of complexity for our model for the mean. Similarly, note that we can apply (3.3.1) or (3.3.2) to covariances created by all manner of wonderful modifications, giving rise to an enormous multitude of possible covariance structures. It wouldn't be unusual, for example, to have a covariance that consists of the sum of a linear term and a periodic term, all multipled by a common modulating amplitude factor. But with this plethora of possibilities for adding terms to our covariance, when we should we stop? In principle, never!

Clearly, complex models include many practical simpler models as subsets. For example, a term in a covariance model can usually be negated by the selection of appropriate hyperparameters - an additive term can have its output scale $\mathbf{h}$ set to zero, a multiplicative term can have its input scale $\mathbf{w}$ set to infinity. It is in selecting the right hyperparameters that Bayesian marginalisation demonstrates its worth. If a term is actively unrepresentative, if it specifies correlations that do not actually exist, the likelihood $p(\, \boldsymbol{y}_D \,|\, \phi,\, I\,)$ will favour those hyperparameters that 'switch off' the misleading term. The likelihood will effectively screen this term from making any contributions to (3.6.1).

Typically, however, we will have a range of possible models that fit the data more or less equally. This wouldn't be a problem, except that they tend to produce different predictions. For such multiple possible models, the argument of William of Occam suggests that we should prefer the simplest. It's common that we might possess a prior that favours the simpler model over a more complex one. However, the remarkable action of Bayesian inference is to penalise an unnecessarily complex model, even if our priors are flat [MacKay, 2002].

In explanation, consider that we wish to compare a complex model C and a simple model S. By this, we mean that the hyperparameter space allowable under C is greater than that allowable under S. For notational covenience we'll define a model variable M that may take either the value C or S. As we're only interested in comparing these two models against each other, we take them to be exhaustive. Moreover, we'll take the flat prior $p(\, M \,|\, I \,) = \frac{1}{2}$. With this, we can rewrite (3.6.1) as

$$p(\, \boldsymbol{x}_\star \,|\, \boldsymbol{y}_D,\, I \,) = \frac{p(\, \boldsymbol{x}_\star \,|\, \boldsymbol{y}_D,\, C,\, I \,)\; p(\, \boldsymbol{y}_D \,|\, C,\, I \,) + p(\, \boldsymbol{x}_\star \,|\, \boldsymbol{y}_D,\, S,\, I \,)\; p(\, \boldsymbol{y}_D \,|\, S,\, I \,)}{p(\, \boldsymbol{y}_D \,|\, C,\, I \,) + p(\, \boldsymbol{y}_D \,|\, S,\, I \,)} \tag{3.6.2}$$

Hence the relevant terms here are the evidences $p(\, \boldsymbol{y}_D \,|\, M,\, I \,)$, whose importance is stressed by Skilling [2006]. These form relative weights for the predictions made under the two models. As such, we're interested in the ratio of their magnitudes

$$\frac{p(\, \boldsymbol{y}_D \,|\, C,\, I \,)}{p(\, \boldsymbol{y}_D \,|\, S,\, I \,)} = \frac{\int p(\, \boldsymbol{y}_D \,|\, \phi,\, C,\, I \,)\; p(\, \phi \,|\, C,\, I \,)\; \mathrm{d}\phi}{\int p(\, \boldsymbol{y}_D \,|\, \phi,\, S,\, I \,)\; p(\, \phi \,|\, S,\, I \,)\; \mathrm{d}\phi} \tag{3.6.3}$$

We take both models to fit the data equally well. That is, there are 'best-fit' configurations of hyperparameters ϕ_M for M such that

$$p(\, \boldsymbol{y}_D \,|\, \phi_C,\, C,\, I \,) = p(\, \boldsymbol{y}_D \,|\, \phi_S,\, S,\, I \,) \tag{3.6.4}$$

To get the flavour of how the inference would proceed, we'll take the Laplace approximation for the integrals in (3.6.3). Hence we assume that both ϕ_M represent the respective solitary peaks of the integrands $p(\, \boldsymbol{y}_D \,|\, \phi,\, M \,)\; p(\, \phi \,|\, M \,)$. These integrands can also be written as $p(\, \phi \,|\, \boldsymbol{y}_D,\, M \,)\; p(\, \boldsymbol{y}_D \,|\, M \,)$ - when integrating over ϕ, the second term is simply a multiplicative constant. Any width of the integrand around its peak is entirely due to $p(\, \phi \,|\, \boldsymbol{y}_D,\, M \,)$. We take measures of these widths as $\Delta_{\phi|D,M}$; such a width would be simply equal to $\sqrt{\det 2\pi K}$ if $p(\, \phi \,|\, \boldsymbol{y}_D,\, M \,)$ were Gaussian with covariance K. Hence we can approximate our integrals as being simply the height times their width of the integrands

$$\begin{aligned}\frac{p(\, \boldsymbol{y}_D \,|\, C,\, I \,)}{p(\, \boldsymbol{y}_D \,|\, S,\, I \,)} &\simeq \frac{p(\, \boldsymbol{y}_D \,|\, \phi_C,\, C,\, I \,)\; p(\, \phi_C \,|\, C,\, I \,)\; \Delta_{\phi|D,C}}{p(\, \boldsymbol{y}_D \,|\, \phi_S,\, S,\, I \,)\; p(\, \phi_S \,|\, S,\, I \,)\; \Delta_{\phi|D,S}} \\ &= \frac{p(\, \phi_C \,|\, C,\, I \,)\; \Delta_{\phi|D,C}}{p(\, \phi_S \,|\, S,\, I \,)\; \Delta_{\phi|D,S}}\end{aligned} \tag{3.6.5}$$

MacKay [2002] calls the terms $p(\, \phi_M \,|\, M,\, I \,)\; \Delta_{\phi|D,M}$ *Occam factors*. For an insight into their meaning, we now further assume that the priors $p(\, \phi \,|\, M,\, I \,)$ are roughly constant over some width $\Delta_{\phi|M}$ and zero elsewhere. Hence

$$\frac{p(\, \boldsymbol{y}_D \,|\, C,\, I \,)}{p(\, \boldsymbol{y}_D \,|\, S,\, I \,)} \simeq \frac{\Delta_{\phi|D,C}}{\Delta_{\phi|C}} \Bigg/ \frac{\Delta_{\phi|D,S}}{\Delta_{\phi|S}} \tag{3.6.6}$$

The width $\Delta_{\phi|D,M}$ is a measure of the sensitivity and flexibility of the model; it represents the volume of hyperparameters consistent with the obtained data. This width is small if the model must be very finely tuned in order to match the data, in which case it will be penalised by the Occam factor. The width $\Delta_{\phi|M}$ is a measure of the complexity of the model; it represents the a priori volume of hyperparameters associated with the model. This width is large if the model has a great number of hyperparameters, in which case it will be penalised by the Occam factor.

Hence we would expect both

$$\Delta_{\phi|D,C} > \Delta_{\phi|D,S} \tag{3.6.7a}$$

$$\Delta_{\phi|C} > \Delta_{\phi|S} \tag{3.6.7b}$$

Hence the Occam factor gives an implicit handicap (3.6.7b) to the complex model, but, if it can justify its flexibility for the received data, it is rewarded (3.6.7a). Effectively, a model will be punished for any 'wasted' hyperparameter space [Gregory, 2005]. The models selected by the marginalisation procedure will have no hyperparameters that are not required to match the data.

The conclusion to be drawn from this is that if we specify a complex model, the machinery inside our Bayesian inference engine will automatically select the simplest sub-model that is sufficiently consistent with the data. In this sense, there is no harm in including as complex and flexible a model as we can dream up - probability theory will do the work for us.

However, the powers of (3.6.1) come at a cost - the integrals are very hard to evaluate! Unfortunately, as demonstrated by (3.5.3) or (3.5.6), the dependence of $p(\, \boldsymbol{x}_\star \mid \boldsymbol{y}_D, \phi, I\,)$ on ϕ is far from simple. ϕ is used to generate means and covariance matrices which are variously inverted, multiplied together and summed, rendering our integrals profoundly nonanalytic. As a consequence, we are forced to use techniques of numerical integration.

Unfortunately, it's well-known that the difficulty of quadrature grows drastically with the dimension of the integral. In our case, this dimension is equal to the number of hyperparameters to be marginalised. As such, this provides a practical limit on the number of terms we can include in our covariance. Each term comes with a new package of hyperparameters, each of which forms a new dimension we must integrate out. Unless our priors over these hyperparameters are very informative, reducing the volume required to be integrated over, they are to be avoided if at all possible. Unless we have reason to believe a term is necessary, it should be left out.

We should nonetheless remain alert to the possibility that the space spanned by our current model is insufficiently broad to include the studied phenomenon, that our model is just too simple to capture the real dynamics. We should be similarly aware when we add covariance terms that can not simply be negated as discussed above, that move rather than simply extend the space spanned by our model.

The quantity that will act as our meter in this regard is again the evidence, $p(\,\boldsymbol{y}_D \mid I\,)$. I here includes the assumptions that underly our selection of a covariance model - given that assigning a prior over models is notoriously difficult and subjective, the model likelihood, represented by the evidence, forms the most informative measure of the efficacy of our model. For exactly the same reasons as discussed above, the evidence will always favour a simpler model if the data is insufficiently discriminatory. If the evidence is small relative to another, we should reconsider our choice of model, its mean and covariance structure. Of course, the evidence forms the denominator of (3.6.1) and hence is a quantity we will necessarily be computing in any case. Skilling [2006] also recommends the publication of the evidence of our final chosen model. This allows any other investigators to objectively compare their methods and models to our own.

Of course, a GP is simply an epistemic probability. Probability theory will ensure that our final results are correct given the specified conditioning information. One consequence of this is that if a (correctly) calculated probability does not reflect our personal beliefs, then only one of two things may be true. Firstly, it is possible that our personal beliefs about the subject are logically inconsistent - our expectations for the calculation were poorly thought out. Otherwise, it must be the case that the model M we fitted our calculation with was not truly representative of our true beliefs. In principle, if we are honestly at all uncertain about M, it should be marginalised. This would entail specifying a prior distribution and then integrating over the space of all possible models - clearly a difficult task. Hence, in practice, we are usually forced to assume a model to work with.

So, to summarise, if a probability seems wrong, either our inputs to or expectations about the output of the inference engine must be wrong. In either case, we have learned something new! This is the essence of how science progresses [Jaynes, 2003] - we trial different assumptions until we find something that seems to fit, even if the resulting theory's predictions appear initially surprising. As John Skilling points out, this is as good as we can ever possibly do. We can never test any theory for all the infinite possible cases to which it should apply, perform every experiment in every possible place and time, to prove it was absolutely 'true'. If we had the truth sitting in front of us, how could we even possibly tell? All we can do is determine the theory that best fits what we've seen so far, and continue to revise that theory as new evidence arrives. This is exactly what is done for us by probability theory.

Hence, if after evaluating the mean and covariance of our GP they do not match our expectations, only one of two things may be possible. Either our expectations or the model we gave the GP were ill-founded.

We are quite within our rights to go back and change the covariance structure and prior distributions until the GP produces results that do match our intuition. Equally, it's possible the GP has revealed a surprising but necessary consequence of our initial information.

3.7 Bayesian Monte Carlo

We return now to the numerical approximation of the marginalisation integrals in (3.6.1). Consider the general challenge of approximating the integral ψ

$$\psi \triangleq \int q(\phi)\, p(\,\phi \,|\, I\,)\, \mathrm{d}\phi \tag{3.7.1}$$

Here we allow $q(\cdot)$ to be any arbitrary function of ϕ - it could be a likelihood, as in the two integrals in (3.6.1), or, indeed any function whose expectation is of interest. (3.7.1) is sufficiently general to describe all integrals required by probability theory. Numerical approximation of (3.7.1) involves the evaluation of the integrand at a set of sample points $\boldsymbol{\phi}_S$. The problem we face is that determining $q(\boldsymbol{\phi}_S)$ is typically a computationally expensive operation. For multidimensional integrals, such as we'll need to perform, the problem is compounded by the 'curse of dimensionality' [Duda et al., 2000]. Essentially, the volume of space required to be mapped out by our samples is exponential in its dimension. This has led to the use of pseudo-random *Monte Carlo* techniques [MacKay, 2002] to circumvent some of these difficulties. Rather than using points chosen uniformly from the whole of ϕ-space, they instead use *sampling* techniques to find regions where the integrand is significant, while still covering as much of ϕ-space as possible. Such techniques have revolutionised Bayesian computation by allowing a practical way to evaluate the high-dimensional integrals so common in real applications.

We now frame the evaluation of (3.7.1) as itself a problem of Bayesian inference [O'Hagan, 1992]. In considering any problem of inference, we need to be clear about both what information we have and which uncertain variables we are interested in. In our case, both the values $q(\boldsymbol{\phi}_S)$ and their locations $\boldsymbol{\phi}_S$ represent valuable pieces of knowledge[4]. However, as we don't know $q(\phi)$ for any $\phi \notin \boldsymbol{\phi}_S$, we are uncertain about the function $q(\cdot)$. As a consequence, we are also uncertain about the value of the integral ψ. As always in the event of uncertainty, this means that we have probability distributions over both $q(\cdot)$ and ψ. The resulting Bayesian network is depicted in Figure 3.5.

But what do we assign to our prior $p(\,q(\cdot)\,|\,I\,)$? Bayesian Monte Carlo [Rasmussen and Ghahramani, 2003], née Bayes-Hermite Quadrature [O'Hagan, 1991] chooses a GP, $p(\,\boldsymbol{q}\,|\,I\,) = \mathbf{N}(\boldsymbol{q};\, \mathbf{0},\, K)$. This will allow us to do regression in an identical fashion to that illustrated in Figure 3.4. We observe the value of our

[4] As discussed out by O'Hagan [1987], traditional, frequentist Monte Carlo effectively ignores the information content of $\boldsymbol{\phi}_S$, leading to several unsatisfactory features.

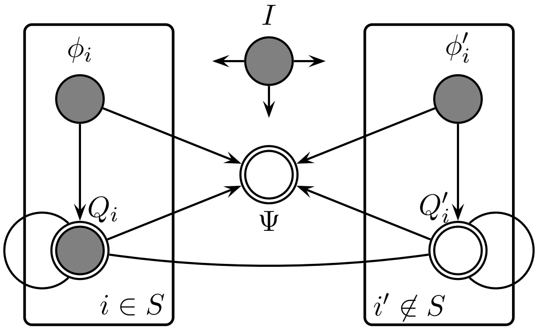


Figure 3.5: Bayesian network for Bayesian Monte Carlo.

function at certain sample locations, use those to infer its value elsewhere and then, with an estimate for the full form of the function, find its integral. As with our convention above, we will take knowledge of sample locations $\boldsymbol{\phi}_S$ to be implicit within I. In line with our treatment of functions as vectors, we will also write $q(\boldsymbol{\phi}_S) = \boldsymbol{q}_S$. Employing this notation, we can use the GP to perform regression for $q(\cdot)$:

$$p(\, \boldsymbol{q} \mid \boldsymbol{q}_S,\, I\,) = \mathbf{N}(\boldsymbol{q};\, \boldsymbol{m},\, \mathbf{C}) \tag{3.7.2}$$

where:

$$\begin{aligned} \boldsymbol{m}_Q(\phi_\star) &\triangleq \mathbf{K}(\phi_\star, \boldsymbol{\phi}_S)\, \mathbf{K}(\boldsymbol{\phi}_S, \boldsymbol{\phi}_S)^{-1}\, \boldsymbol{q}_S \\ \mathbf{C}_Q(\phi_\star, \phi'_\star) &\triangleq \mathbf{K}(\phi_\star, \phi'_\star) - \mathbf{K}(\phi_\star, \boldsymbol{\phi}_S)\, \mathbf{K}(\boldsymbol{\phi}_S, \boldsymbol{\phi}_S)^{-1}\, \mathbf{K}(\boldsymbol{\phi}_S, \phi'_\star) \end{aligned} \tag{3.7.3}$$

Given this distribution for $q(\cdot)$, we can now employ probability theory to find the distribution for ψ

$$\begin{aligned} p(\, \psi \mid \boldsymbol{q}_S,\, I\,) &= \int p(\, \psi \mid \boldsymbol{q},\, I\,)\; p(\, \boldsymbol{q} \mid \boldsymbol{q}_S,\, I\,)\; \mathrm{d}\boldsymbol{q} \\ &= \int \delta\Big(\psi - \int p(\, \phi_\star \mid I\,)\; q\, \mathrm{d}\phi_\star\Big)\; \mathbf{N}(q;\, m_Q,\, C_Q)\; \mathrm{d}q \\ &= \lim_{\varepsilon \to 0} \int \mathbf{N}\Big(\psi;\, \int p(\, \phi_\star \mid I\,)\; q\, \mathrm{d}\phi_\star,\, \varepsilon\Big)\; \mathbf{N}(q;\, m_Q,\, C_Q)\; \mathrm{d}q \\ &= \mathbf{N}(\psi;\, m_\Psi,\, C_\Psi) \end{aligned} \tag{3.7.4}$$

where, noting that the integral operator $\int \mathrm{d}\phi_\star\, p(\, \phi_\star \mid I\,)$ is a projection, we may use (A.1.7) to write

$$\begin{aligned} m_\Psi &\triangleq \int p(\, \phi_\star \mid I\,)\; m_Q(\phi_\star)\, \mathrm{d}\phi_\star \\ &= \mathfrak{n}_S^{\mathrm{T}}\, \mathbf{K}(\boldsymbol{\phi}_S, \boldsymbol{\phi}_S)^{-1}\, \boldsymbol{q}_S \end{aligned} \tag{3.7.5}$$

$$\begin{aligned} C_\Psi &\triangleq \iint p(\, \phi_\star \mid I\,)\; C_Q(\phi_\star, \phi'_\star)\, p(\, \phi'_\star \mid I\,)\; \mathrm{d}\phi_\star \mathrm{d}\phi'_\star \\ &= \iint p(\, \phi_\star \mid I\,)\; K(\phi_\star, \phi'_\star)\, p(\, \phi'_\star \mid I\,)\; \mathrm{d}\phi_\star \mathrm{d}\phi'_\star - \mathfrak{n}_S^{\mathrm{T}}\, \mathbf{K}(\boldsymbol{\phi}_S, \boldsymbol{\phi}_S)^{-1} \mathfrak{n}_S \end{aligned} \tag{3.7.6}$$

for

$$\mathfrak{n}_S \triangleq \int p(\, \phi_\star \mid I\,)\; \mathbf{K}(\phi_\star, \boldsymbol{\phi}_S)\, \mathrm{d}\phi_\star \tag{3.7.7}$$

To progress further, we assume a squared exponential covariance function and multivariate Gaussian prior on ϕ

$$p(\,\phi \,|\, I\,) \triangleq \mathbf{N}\big(\phi;\, \boldsymbol{\nu},\, \lambda^{\mathrm{T}}\lambda\big) \tag{3.7.8}$$

$$K(\phi, \phi') \triangleq h^2\,\mathbf{N}\big(\phi;\, \phi',\, \mathbf{w}^{\mathrm{T}}\mathbf{w}\big) \tag{3.7.9}$$

noting that here h is not quite the output scale, due to the normalisation constant of the Gaussian. We now have

$$\iint p(\,\phi_\star \,|\, I\,)\; K(\phi_\star, \phi'_\star)\, p(\,\phi'_\star \,|\, I\,)\; \mathrm{d}\phi_\star \mathrm{d}\phi'_\star = \frac{h^2}{\sqrt{\det 2\pi(\lambda^{\mathrm{T}}\lambda + \mathbf{w}^{\mathrm{T}}\mathbf{w})}} \tag{3.7.10}$$

$$\mathfrak{n}_S(i) = h^2\,\mathbf{N}\big(\phi_i;\, \boldsymbol{\nu},\, \lambda^{\mathrm{T}}\lambda + \mathbf{w}^{\mathrm{T}}\mathbf{w}\big)\,; \quad \forall i \in S \tag{3.7.11}$$

Hence, finally, the mean and covariance of our Gaussian for Ψ are

$$m_\Psi = \mathfrak{n}_S^{\mathrm{T}}\,\mathbf{K}(\boldsymbol{\phi}_S, \boldsymbol{\phi}_S)^{-1}\,\boldsymbol{q}_S \tag{3.7.12}$$

$$C_\Psi = \frac{h^2}{\sqrt{\det 2\pi(\lambda^{\mathrm{T}}\lambda + \mathbf{w}^{\mathrm{T}}\mathbf{w})}} - \mathfrak{n}_S^{\mathrm{T}}\,\mathbf{K}(\boldsymbol{\phi}_S, \boldsymbol{\phi}_S)^{-1}\mathfrak{n}_S \tag{3.7.13}$$

Note that the form of (3.7.12), a linear combination of the samples q_i, is identical to that of traditional quadrature or Monte Carlo techniques.

Of course, if $q(\phi)$ is a likelihood $p(\,x \,|\, \phi,\, I\,)$, ψ will be a probability distribution itself. In this case, we have a second-order probability, as discussed in Section 2.2, for ψ. As such, we can write

$$\psi = p(\,x \,|\, \boldsymbol{q}_S,\, \boldsymbol{q}_{\overline{S}},\, I\,) \tag{3.7.14}$$

where $\boldsymbol{q}_{\overline{S}}$ represents the complement of $\boldsymbol{q}_S$; together, they represent all possible values of the function $q(\cdot)$. $\boldsymbol{q}_{\overline{S}}$ forms the additional information J we considered in Section 2.2. Of course, we don't actually know $\boldsymbol{q}_{\overline{S}}$, so the quantity of interest is, as before

$$\begin{aligned} p(\,x \,|\, \boldsymbol{q}_S,\, I\,) &= \iint p(\,x \,|\, \boldsymbol{q}_S,\, \boldsymbol{q}_{\overline{S}},\, I\,)\; p(\,\psi \,|\, \boldsymbol{q}_S,\, \boldsymbol{q}_{\overline{S}},\, I\,)\; p(\,\boldsymbol{q}_{\overline{S}} \,|\, \boldsymbol{q}_S,\, I\,)\; \mathrm{d}\boldsymbol{q}_{\overline{S}}\,\mathrm{d}\psi \\ &= \int p(\,\psi \,|\, \boldsymbol{q}_S,\, I\,)\; \mathrm{d}\psi \end{aligned} \tag{3.7.15}$$

Hence inference about x is only ever dependent on the mean of $p(\,\psi \,|\, S,\, I\,)$, m_Ψ. Its higher moments are of interest only inasmuch as we consider how our beliefs about ψ would change if we had taken different sets of samples $\boldsymbol{\phi}_S$.

This fact leads to the immediate conclusion that the choice of $\mathbf{h}$ is irrelevant to inference about x. Note that h^2 simply drops out of the calculation of (3.7.12) above. Indeed, this is true for the mean of any GP for a noiseless process. This is fortunate, after all in our case, $\mathbf{h}$ forms a hyper-hyperparameter, a parameter of the covariance expressing how expect q to vary with ϕ. Practically, having to marginalise such hyper-hyperparameters is not a particularly appealing prospect. Unfortunately, we still have to deal with $\mathbf{w}$, which for this reason of practicality we'll need to set to a fixed value. One approach is to assume that the scales of variation of $q(\phi)$ will be the same as that for the prior, giving $\mathbf{w}^{\mathrm{T}}\mathbf{w}$ equal to the covariance of $p(\ \phi \mid I\ )$. In any case, fortunately our inference is not particularly sensitive to the selection of this hyper-hyperparameter. If we take a reasonable number of samples, the GP will be so constrained by them that its beliefs about how q varies elsewhere will not prove greatly significant.

One particular weakness of the approach taken above is that the GP can not be readily informed of the non-negativity of $q(\phi)$, in the case that it represents a likelihood. Of course, it would be preferable to place a GP on the logarithm of $q(\phi)$. Unfortunately, this would ruin our ability to perform the integral leading to (3.7.4) by making the transformation of q non-linear. Fortunately, this weakness is not terminal. Again, so long as the GP is given sufficient samples in the region of interest, it will learn the non-negativity there simply from the weight of data. Outside of the sampled region, the posterior mean of the GP will remain at its prior mean, which we have set as zero.

We now return to what may have appeared another weak link in the reasoning above, our assumption of a Gaussian prior (3.7.8) for ϕ. Clearly this assumption was motivated by our desire for analytic solutions to the integrals in (3.7.6) and (3.7.6). Note that these integrals will be equally analytic if we had taken a log-normal prior and a squared exponential covariance in the log of ϕ. This offers a neat solution for hyperparameters restricted to the positive reals. Indeed, if there is any transform of our hyperparameter upon which we are happy in placing a normal prior and squared exponential covariance, our integrals can be resolved in an identical way.

However, we have a still greater degree of flexibility. Any prior that can be analytically integrated with respect to some concoction of covariance function is admissable. Examples include Gaussian mixture models and polynomial kernels. Of course, even this may prove unsatisfactory - choosing a prior for its analytic properties is an approximation that may not always be justifiable. Choosing a covariance function, representing our beliefs about how q changes with ϕ, purely to match a particular prior forms a similar misuse of probability theory. Take then the very worst case scenario, in which our prior is not at all analytic with respect to the covariance. As a possible solution, note that the separation of the prior and q was only performed at all in order to try and resolve (3.7.6) and (3.7.7). If this has failed, perhaps there is another multiplicative component of the integrand $q(\phi)\, p(\ \phi \mid I\ )$ which can be broken off to fill the role taken by the prior above.

If even this fails, we could try treat the integrand as a single indivisable entity, incorporating $p(\ \phi \mid I\ )$ into $q(\phi)$. This yields $\mathfrak{n}_S = h^2\,\mathbf{1}$ and hence a reasonable m_Ψ, but infinite C_Ψ, for any isotropic and stationary covariance. Such covariance functions represent the only sensible choice unless we have exceptional knowledge about how the integrand varies with ϕ. To understand what the GP is telling us here, note that effectively what we have done is to take a flat, improper prior for ϕ. Hence regardless of the number of ϕ samples we supply it with, the GP can never be at all sure that it has caught all regions where the integrand is significant. We have told it that ϕ space stretches out to infinity in all directions - as far as it knows, the most important lump of integrand mass could be infinitely far removed from the areas it has been told about so far. Given that the correlations the GP assumes hold only between points at a few length scales remove, it has done exactly the right thing in doing its best to report an estimate but warning us that it could be infinitely wrong. The conclusion to draw is that we must employ some restrictive information about the integrand or else be literally infinitely uncertain.

We now turn to the matter of where the samples $\boldsymbol{\phi}_S$ actually come from. Note that the GP will perform the correct regression for $q(\phi)$ no matter how the samples are obtained. As such, BMC can employ any of the myriad of proposed sampling schemes, as reviewed in Neal [1993], Loredo [1999] and MacKay [2002]. Modern advances include Nested Sampling [Skilling, 2006], Parallel Tempering [Gregory, 2005, Chapter 12] and using GPs for Hamiltonian (aka Hybrid) Monte Carlo (HMC) [Rasmussen, 2003].

3.8 Marginalising Revisited

Given the techniques of BMC, we can now revisit our marginalisation integrals (3.6.1). This problem is a little more sophisticated than the generic integration problem we have considered above. In particular, there are two functions we are uncertain about

$$q(\phi) \triangleq p(\, \boldsymbol{x}_\star \mid \boldsymbol{y}_D,\, \phi,\, I\,) \tag{3.8.1}$$

$$r(\phi) \triangleq p(\, \boldsymbol{y}_D \mid \phi,\, I\,) \tag{3.8.2}$$

In our case, $r(\phi)$ appears in both our numerator and denominator integrals, introducing correlations between the values we estimate for them. The correlation structure of this system is illustrated in Figure 3.6.

As such, it is strictly incorrect to treat the two integrals independently using the methods proposed above. Instead, we are interested in the full quantity

$$\psi \triangleq \frac{\int q\, r\, p(\, \phi_\star \mid I\,)\, \mathrm{d}\phi_\star}{\int r\, p(\, \phi_\star \mid I\,)\, \mathrm{d}\phi_\star} \tag{3.8.3}$$

From above, recall that it is only the mean of $p(\, \psi \mid \boldsymbol{q}_S,\, \boldsymbol{r}_S,\, I\,)$ that will ever enter our inference about $\boldsymbol{x}_\star$. Hence we can limit ourselves to its calculation, defining the GP parameters m_Q, C_Q, m_R and C_R to be the

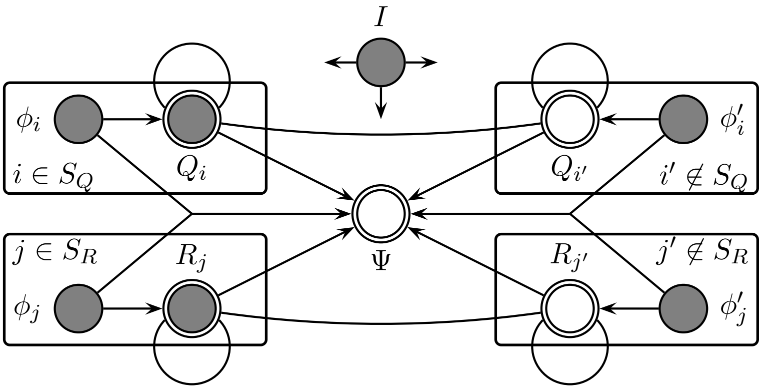


Figure 3.6: Bayesian network for marginalising hyperparameters using BMC.

expected regression equations as in (3.7.3),

$$\begin{aligned} m_\Psi &\triangleq \int \psi\, p(\,\psi \mid \boldsymbol{q}_S,\, \boldsymbol{r}_S,\, I\,)\, \mathrm{d}\psi \\ &= \iiint \psi\, p(\,\psi \mid \boldsymbol{q},\, \boldsymbol{r},\, I\,)\, p(\,\boldsymbol{q} \mid \boldsymbol{q}_S,\, I\,)\, p(\,\boldsymbol{r} \mid \boldsymbol{r}_S,\, I\,)\, \mathrm{d}\psi\, \mathrm{d}\boldsymbol{q}\, \mathrm{d}\boldsymbol{r} \\ &= \iiint \psi\, \delta\!\left(\psi - \frac{\int q\, r\, p(\,\phi_\star \mid I\,)\, \mathrm{d}\phi_\star}{\int r\, p(\,\phi_\star \mid I\,)\, \mathrm{d}\phi_\star}\right) \mathbf{N}(q;\, m_Q,\, C_Q)\, \mathbf{N}(r;\, m_R,\, C_R)\, \mathrm{d}\psi\, \mathrm{d}q\, \mathrm{d}r \\ &= \int \frac{\int m_Q\, r\, p(\,\phi_\star \mid I\,)\, \mathrm{d}\phi_\star}{\int r\, p(\,\phi_\star \mid I\,)\, \mathrm{d}\phi_\star}\, \mathbf{N}(r;\, m_R,\, C_R)\, \mathrm{d}r \end{aligned} \tag{3.8.4}$$

Here, unfortunately, our integration over r becomes nonanalytic. However, we can employ the Laplace approximation by expanding the integrand around an assumed peak at m_R. We will additionally assume that our uncertainty C_R is small relative to the mean m_R. This is a fairly weak assumption - after all, the computation of the likelihoods r_S is a near noiseless process. Any uncertainty that exists will be only in the value of the likelihood away from our sample set. If we take a sufficient number of samples, we can justifiably take the mean of our GP as well representative of the function $r(\cdot)$. Hence we can use the approximation

$$\det(\mathbf{I} + \mathbf{M})^{-\frac{1}{2}} \simeq \big(1 + \mathrm{tr}(\mathbf{M})\big)^{-\frac{1}{2}} \simeq 1 - \frac{1}{2}\,\mathrm{tr}(\mathbf{M}) \tag{3.8.5}$$

which holds for a matrix $\mathbf{M}$ whose entries are small relative to the identity. This gives

$$\begin{aligned} m_\Psi \simeq \frac{\int m_Q(\phi_\star)\, m_R(\phi_\star)\, p(\,\phi_\star \mid I\,)\, \mathrm{d}\phi_\star}{\int m_R(\phi_\star)\, p(\,\phi_\star \mid I\,)\, \mathrm{d}\phi_\star} \Bigg(1 &+ \frac{\iint p(\,\phi_\star \mid I\,)\, C_R(\phi_\star, \phi'_\star)\, p(\,\phi'_\star \mid I\,)\, \mathrm{d}\phi_\star\, \mathrm{d}\phi'_\star}{2\,(\int m_R(\phi_\star)\, p(\,\phi_\star \mid I\,)\, \mathrm{d}\phi_\star)^2} \\ &- \frac{\iint p(\,\phi_\star \mid I\,)\, m_Q(\phi_\star)\, C_R(\phi_\star, \phi'_\star)\, m_Q(\phi'_\star)\, p(\,\phi'_\star \mid I\,)\, \mathrm{d}\phi_\star\, \mathrm{d}\phi'_\star}{2\,(\int m_Q(\phi_\star)\, m_R(\phi_\star)\, p(\,\phi_\star \mid I\,)\, \mathrm{d}\phi_\star)^2} \Bigg) \end{aligned} \tag{3.8.6}$$

If we take a Gaussian prior and squared exponential covariance as in (3.7.8) and (3.7.9), and noting that $\mathbf{K}$ over $\boldsymbol{\phi}_{S,Q}$ is understood to be potentially generated by a different covariance function to $\mathbf{K}$ over $\boldsymbol{\phi}_{S,R}$, we have

$$\int p(\,\phi_\star \mid I\,)\, \mathbf{K}(\phi_\star, \boldsymbol{\phi}_{S,R})\, \mathrm{d}\phi_\star = \mathfrak{n}_{S,R} \tag{3.8.7}$$

$$\int \mathbf{K}(\boldsymbol{\phi}_{S,Q}, \phi_\star)\, p(\,\phi_\star \mid I\,)\, \mathbf{K}(\phi_\star, \boldsymbol{\phi}_{S,R})\, \mathrm{d}\phi_\star = \mathfrak{N}_S \tag{3.8.8}$$

where we define, $\forall i \in S_Q,\ \forall j \in S_R$

$$\mathfrak{n}_{S,R}(j) \triangleq h_R^2\, \mathbf{N}\big(\phi_j;\, \boldsymbol{\nu},\, \lambda^{\mathrm{T}}\lambda + \mathbf{w}_R^{\mathrm{T}}\mathbf{w}_R\big) \tag{3.8.9}$$

$$\mathfrak{N}_S(i,j) \triangleq h_Q^2\, h_R^2\, \mathbf{N}\left(\begin{bmatrix}\boldsymbol{\phi}_i\\ \boldsymbol{\phi}_j\end{bmatrix};\, \begin{bmatrix}\boldsymbol{\nu}\\ \boldsymbol{\nu}\end{bmatrix},\, \begin{bmatrix}\lambda^{\mathrm{T}}\lambda + \mathbf{w}_Q^{\mathrm{T}}\mathbf{w}_Q & \lambda^{\mathrm{T}}\lambda\\ \lambda^{\mathrm{T}}\lambda & \lambda^{\mathrm{T}}\lambda + \mathbf{w}_R^{\mathrm{T}}\mathbf{w}_R\end{bmatrix}\right) \tag{3.8.10}$$

and so

$$m_\Psi \simeq \frac{\boldsymbol{q}_S^{\mathrm{T}}\, \mathbf{K}(\boldsymbol{\phi}_{S,Q}, \boldsymbol{\phi}_{S,Q})^{-1}\, \mathfrak{N}_S\, \mathbf{K}(\boldsymbol{\phi}_{S,R}, \boldsymbol{\phi}_{S,R})^{-1}\, \boldsymbol{r}_S}{\mathfrak{n}_{S,R}^{\mathrm{T}}\, \mathbf{K}(\boldsymbol{\phi}_{S,R}, \boldsymbol{\phi}_{S,R})^{-1}\, \boldsymbol{r}_S}\, (1 + \ldots) \tag{3.8.11}$$

Note that (3.8.11) can be viewed as a linear combination of the elements of $\boldsymbol{q}_S$, just as we found in Section 3.7. In other words, the mean of our integral is a weighted mixture of the samples $q_i = p(\, \boldsymbol{x}_\star \mid \phi_i,\, \boldsymbol{y}_D,\, I\,)$

$$m_\Psi = \boldsymbol{q}_S^{\mathrm{T}}\, \boldsymbol{\rho}'\, \kappa \tag{3.8.12}$$

where the weights are given by the product of the two terms

$$\boldsymbol{\rho}' \triangleq \mathbf{K}(\boldsymbol{\phi}_{S,Q}, \boldsymbol{\phi}_{S,Q})^{-1}\, \mathfrak{N}_S\, \mathbf{K}(\boldsymbol{\phi}_{S,R}, \boldsymbol{\phi}_{S,R})^{-1}\, \boldsymbol{r}_S \tag{3.8.13}$$

$$\kappa \triangleq \frac{1}{\mathfrak{n}_{S,R}^{\mathrm{T}}\, \mathbf{K}(\boldsymbol{\phi}_{S,R}, \boldsymbol{\phi}_{S,R})^{-1}\, \boldsymbol{r}_S}\, (1 + \ldots) \tag{3.8.14}$$

All elements within the multiplicative bracketed term in (3.8.14) are dimensionless constants; κ represents only a rescaling of the weights, not an alteration of their ratios. Note that, given that $m_\Psi = p(\, \boldsymbol{x}_\star \mid \boldsymbol{y}_D,\, \boldsymbol{q}_S,\, \boldsymbol{r}_S,\, I\,)$ and each individual q_i must all be valid probability distributions for $\boldsymbol{x}_\star$,

$$1 = \int m_\Psi\, \mathrm{d}\boldsymbol{x}_\star = \int \left(\sum_i q_i\, \rho'_i\, \kappa\right) \mathrm{d}\boldsymbol{x}_\star = \sum_i \left(\int q_i\, \mathrm{d}\boldsymbol{x}_\star\right) \rho'_i\, \kappa = \kappa \sum_i \rho'_i \tag{3.8.15}$$

The requirement that (3.8.12) be a valid probability distribution forces the weights to be normalised. As such, we need not actually compute (3.8.14) - if we calculate ρ', clearly then $\kappa = (\sum_i \rho'_i)^{-1}$. Hence we define

$$\boldsymbol{\rho} \triangleq \boldsymbol{\rho}'\, \kappa = \frac{\mathbf{K}(\boldsymbol{\phi}_{S,Q}, \boldsymbol{\phi}_{S,Q})^{-1}\, \mathfrak{N}_S\, \mathbf{K}(\boldsymbol{\phi}_{S,R}, \boldsymbol{\phi}_{S,R})^{-1}\, \boldsymbol{r}_S}{\mathbf{1}_{S,1}^{\mathrm{T}}\, \mathbf{K}(\boldsymbol{\phi}_{S,Q}, \boldsymbol{\phi}_{S,Q})^{-1}\, \mathfrak{N}_S\, \mathbf{K}(\boldsymbol{\phi}_{S,R}, \boldsymbol{\phi}_{S,R})^{-1}\, \boldsymbol{r}_S} \tag{3.8.16}$$

where $\mathbf{1}_{S,1}$ is a vector containing only ones of dimensions equal to $\boldsymbol{q}_S$. Expressed in terms of (3.8.16), our final result is

$$m_\Psi = \boldsymbol{q}_S^{\mathrm{T}}\, \boldsymbol{\rho} \tag{3.8.17}$$

Note this is identical to the result obtained by taking C_R as exactly zero. Effectively, so long as this uncertainty is not large enough to disrupt our assumption that m_Ψ is a linear combination of the q_i's, (3.8.16) and (3.8.17) will hold. Note that if we place a GP on $p(\, \boldsymbol{x}_\star \mid \phi,\, I\,)$ and assume Gaussian noise, then each $q_i = p(\, \boldsymbol{x}_\star \mid \boldsymbol{y}_D,\, \phi_i,\, I\,)$ will be a slightly different Gaussian. We define the resulting mean and covariance of each q_i as $\boldsymbol{\mu}_{\star,i}$ and $\boldsymbol{\Lambda}_{\star,i}$ respectively. Hence m_Ψ is a Gaussian (process) mixture. As such, we can express its mean and covariance as

$$\boldsymbol{\mu}_\star \triangleq \int \boldsymbol{x}_\star \, p(\, \boldsymbol{x}_\star \mid \boldsymbol{y}_D, \, \phi_i, \, I \,) \, \mathrm{d}\boldsymbol{x}_\star = \sum_{i \in S_Q} \rho_i \, \boldsymbol{\mu}_{\star,i} \tag{3.8.18}$$

$$\begin{aligned} \boldsymbol{\Lambda}_\star &\triangleq \int \boldsymbol{x}_\star \, \boldsymbol{x}_\star^{\mathrm{T}} \, p(\, \boldsymbol{x}_\star \mid \boldsymbol{y}_D, \, \phi_i, \, I \,) \, \mathrm{d}\boldsymbol{x}_\star - \boldsymbol{\mu}_\star \, \boldsymbol{\mu}_\star^{\mathrm{T}} \\ &= \sum_{i \in S_Q} \rho_i \, \left(\boldsymbol{\Lambda}_{\star,i} + \boldsymbol{\mu}_{\star,i} \, \boldsymbol{\mu}_{\star,i}^{\mathrm{T}} \right) - \boldsymbol{\mu}_\star \, \boldsymbol{\mu}_\star^{\mathrm{T}} \end{aligned} \tag{3.8.19}$$

Chapter 4

Iterative methods

4.1 GP Updating

In real problems of prediction, we do not have a fixed data set D. Rather, we receive data one or a few observations at a time, the total information available to us perpetually growing incrementally. Moreover, updates are often required every hour, minute or even second - time is critical. Using the above methods as is for this case would rapidly incur prohibitive computational overheads. Each time we wish to update our beliefs in the light of a new observation, we would be forced to perform the expensive Cholesky decomposition (the complexity of which grows as $O(\frac{1}{3}n^3)$ in the dimension n of the matrix) of an ever-bigger $\mathbf{V}$ matrix. Fortunately, there are a number of tricks we can employ to improve the efficiency of the iterative updates.

The following proposals all exploit the special structure of our problem. When we receive new data, our $\mathbf{V}$ matrix is changed only in the addition of a few new rows and columns. Hence most of the work that went into computing its Cholesky decomposition at the last iteration can be recycled, if we are careful. To reflect the iterative nature of our data, we denote by D_n the data obtained at the nth iteration, and by $D_{1:n}$ all the data obtained between the first and nth iterations, inclusively. Where used as an argument or subscript, the D may also be dropped, if clear.

The first problem we face is thus: we know $\mathbf{R}(\boldsymbol{t}_{1:n-1}, \boldsymbol{t}_{1:n-1})$ and wish to compute $\mathbf{R}(\boldsymbol{t}_{1:n}, \boldsymbol{t}_{1:n})$. Clearly

$$\begin{bmatrix} \mathbf{V}(\boldsymbol{t}_{1:n-1}, \boldsymbol{t}_{1:n-1}) & \mathbf{V}(\boldsymbol{t}_{1:n-1}, \boldsymbol{t}_n) \\ \mathbf{V}(\boldsymbol{t}_n, \boldsymbol{t}_{1:n-1}) & \mathbf{V}(\boldsymbol{t}_n, \boldsymbol{t}_n) \end{bmatrix} = \mathbf{V}(\boldsymbol{t}_{1:n}, \boldsymbol{t}_{1:n}) \tag{4.1.1}$$

We also know that $\mathbf{R}(\boldsymbol{t}_{1:n}, \boldsymbol{t}_{1:n})$ must have the form

$$\mathbf{R}(\boldsymbol{t}_{1:n}, \boldsymbol{t}_{1:n}) = \begin{bmatrix} \mathbf{R}(\boldsymbol{t}_{1:n-1}, \boldsymbol{t}_{1:n-1}) & \mathbf{S} \\ 0 & \mathbf{U} \end{bmatrix} \tag{4.1.2}$$

for some $\mathbf{S}$ and upper triangular $\mathbf{U}$, as can be seen by rewriting (4.1.1) as

$$\begin{aligned}\begin{bmatrix} \mathbf{R}(\boldsymbol{t}_{1:n-1},\boldsymbol{t}_{1:n-1})^{\mathrm{T}}\,\mathbf{R}(\boldsymbol{t}_{1:n-1},\boldsymbol{t}_{1:n-1}) & \mathbf{V}(\boldsymbol{t}_{1:n-1},\boldsymbol{t}_n) \\ \mathbf{V}(\boldsymbol{t}_n,\boldsymbol{t}_{1:n-1}) & \mathbf{V}(\boldsymbol{t}_n,\boldsymbol{t}_n) \end{bmatrix} &= \mathbf{R}(\boldsymbol{t}_{1:n},\boldsymbol{t}_{1:n})^{\mathrm{T}}\,\mathbf{R}(\boldsymbol{t}_{1:n},\boldsymbol{t}_{1:n}) \\ &= \begin{bmatrix} \mathbf{R}(\boldsymbol{t}_{1:n-1},\boldsymbol{t}_{1:n-1})^{\mathrm{T}}\,\mathbf{R}(\boldsymbol{t}_{1:n-1},\boldsymbol{t}_{1:n-1}) & \mathbf{R}(\boldsymbol{t}_{1:n-1},\boldsymbol{t}_{1:n-1})^{\mathrm{T}}\,\mathbf{S} \\ \mathbf{S}^{\mathrm{T}}\,\mathbf{R}(\boldsymbol{t}_{1:n-1},\boldsymbol{t}_{1:n-1}) & \mathbf{S}^{\mathrm{T}}\,\mathbf{S}+\mathbf{U}^{\mathrm{T}}\,\mathbf{U} \end{bmatrix} \end{aligned} \tag{4.1.3}$$

and noting that the Cholesky decomposition is unique for a positive definite matrix. Hence it can be seen that

$$\mathbf{S} = \mathbf{R}(\boldsymbol{t}_{1:n-1},\boldsymbol{t}_{1:n-1})^{\mathrm{T}} \setminus \mathbf{V}(\boldsymbol{t}_{1:n-1},\boldsymbol{t}_n) \tag{4.1.4}$$

$$\mathbf{U} = \operatorname{chol}\big(\mathbf{V}(\boldsymbol{t}_n,\boldsymbol{t}_n) - \mathbf{S}^{\mathrm{T}}\,\mathbf{S}\big) \tag{4.1.5}$$

Hence this provides an efficient method to update our Cholesky factor as new data is obtained. It's worth noting the equivalent method of updating a matrix inverse is given by use of the *inversion by partitioning* formulae [Press et al., 1992, Section 2.7].

An efficient updating rule for $\mathfrak{a}$ also exists

$$\begin{aligned} \mathfrak{a}_{1:n} &= \mathbf{R}(\boldsymbol{t}_{1:n},\boldsymbol{t}_{1:n})^{\mathrm{T},-1}\,(\boldsymbol{y}_{1:n} - \mu(\boldsymbol{t}_{1:n})) \\ &= \begin{bmatrix} \mathbf{R}(\boldsymbol{t}_{1:n-1},\boldsymbol{t}_{1:n-1})^{\mathrm{T},-1} & 0 \\ -\mathbf{U}^{\mathrm{T},-1}\,\mathbf{S}^{\mathrm{T}}\,\mathbf{R}(\boldsymbol{t}_{1:n-1},\boldsymbol{t}_{1:n-1})^{\mathrm{T},-1} & \mathbf{U}^{\mathrm{T},-1} \end{bmatrix} (\boldsymbol{y}_{1:n} - \mu(\boldsymbol{t}_{1:n})) \\ &= \begin{bmatrix} \mathfrak{a}_{1:n-1} \\ \mathfrak{c}_i \end{bmatrix} \end{aligned} \tag{4.1.6}$$

where

$$\mathfrak{c}_i \triangleq \mathbf{U}^{\mathrm{T}} \setminus \big(\boldsymbol{y}_i - \mu(\boldsymbol{t}_n) - \mathbf{S}^{\mathrm{T}}\,\mathfrak{a}_{1:n-1}\big) \tag{4.1.7}$$

Note that in a prediction context, $\mathfrak{b}_{\star,D}$ is not suitable for such rules. This is due to its dependence on the selection of predictants, denoted by the $\star$. In prediction and tracking, the objects of our interest will be constantly shifting, implying no recycling is possible.

So, after all these update rules, it's time to weigh the ledger to determine just what we have gained. The most expensive operations above, such as the determination of $\mathbf{S}$, (4.1.4), and $\mathfrak{b}_{\star,D}$, (3.5.5), involve the solution of a set of n triangular equations at time step n, the cost of which will in general scale as $O(n^2)$. As such, we have reduced the overall cost of an update from $O(n^3)$ to $O(n^2)$, with no loss of accuracy.

4.2 Iterative Marginalisation

We now reconsider our marginalisation procedure. Ideally, in order to evaluate the integrals in (3.6.1), we would like to sample ϕ from the whole integrands $p(\,\boldsymbol{x}_\star,\,\boldsymbol{y}_D,\,\phi\,|\,I\,)$ and $p(\,\boldsymbol{y}_D,\,\phi\,|\,I\,)$ for numerator and denominator respectively. For this, we could employ any of the sophisticated sampling techniques mentioned in Section 3.7. However, when it comes to our iterative procedure, we need to perform this marginalisation at each time step, as $\boldsymbol{y}_D$ and $\boldsymbol{x}_\star$ are constantly changing. It's questionable whether we can afford to perform a computationally expensive sampling from our changing integrands every time we wish to update our beliefs. Also of importance is that the update rules we have just described in Section 4.1 hold only for a fixed set of hyperparameters ϕ_i. We can't use computations performed on one ϕ_i in order to update another. Hence, if we resample at each time step, we'll have to perform all our GP computation from scratch. For this reason, we rule out such procedures for now.

Given this prohibition, for now we take the simplest possible approach. We sample our hyperparameters only once, off-line, and maintain these samples throughout. This will allow us to use all our update rules for each sample ϕ_i separately, giving us effectively a GP running in parallel for each one. These will then be combined in the Gaussian mixture (3.8.17) to give our final estimates at each time step.

Note that the only computations that are expensive in the number η of samples are the Cholesky decomposition and multiplication of covariance matrices in (3.8.16). These will scale as $O(\eta^3)$. Otherwise, the evaluation of our Gaussian mixture (3.8.18) and (3.8.19) scale only linearly with η. Fortunately, if we use a fixed set of samples as just proposed, the problematic term $\mathbf{K}_Q^{-1}\,\mathfrak{N}_S\,\mathbf{K}_R^{-1}$ in (3.8.16) need only be evaluated once, off-line. This is an important boon, rendering it feasible for us to consider the large number of hyperparameters necessary to effectively explore our multi-dimensional integrals.

Further simplications can be achieved by using only a single set of samples $\boldsymbol{\phi}_S$ for both integrals. If $\boldsymbol{\phi}_S$ forms a good coverage of the integrand $r(\phi_\star)\,p(\,\phi_\star \mid I\,)$, it's fair to assume that it will equally form a reasonable guide to where $q(\phi_\star)\,r(\phi_\star)\,p(\,\phi_\star \mid I\,)$ is significant. We may also use the same $\mathbf{w}$ in the squared exponential covariances for both q and r, expressing that we expect them both to vary over ϕ at identical scales. These choices mean that we must evaluate and factorise only a single covariance matrix[1] in (3.8.16).

Yet another substantial saving is found if we are willing to assign independent Gaussian priors to each hyperparameter $\phi_{(e)} \in \phi$. We may also be willing to couple this with a covariance structure of the form

$$K(\phi,\,\phi') = \prod_e K_e(\phi_{(e)},\,\phi'_{(e)}) \qquad (4.2.1)$$

so that the correlations are likewise an independent product of terms over each hyperparameter. If we additionally consider a simple grid of samples, such that $\boldsymbol{\phi}_S$ is the tensor product of a set of samples $\boldsymbol{\phi}_{(e),S}$ over each

[1] Strictly, the covariances for q and r are different in their output scales h, as they must encode the different dimensionalities of r, a function of $\boldsymbol{y}_D$, and q, additionally a function of $\boldsymbol{x}_\star$. Aside from this simple multiplicative constant, the two covariance matrices are equal.

hyperparameter, then we have

$$\begin{aligned}
&\mathbf{K}(\boldsymbol{\phi}_{S,Q},\boldsymbol{\phi}_{S,Q})^{-1}\,\mathfrak{N}_S\,\mathbf{K}(\boldsymbol{\phi}_{S,R},\boldsymbol{\phi}_{S,R})^{-1} = \\
&\qquad \mathbf{K}(\boldsymbol{\phi}_{(1),Q,S},\boldsymbol{\phi}_{(1),Q,S})^{-1}\,\mathfrak{N}_S(\boldsymbol{\phi}_{(1),Q,S},\boldsymbol{\phi}_{(1),R,S})\,\mathbf{K}(\boldsymbol{\phi}_{(1),R,S},\boldsymbol{\phi}_{(1),R,S})^{-1} \\
&\quad \otimes \mathbf{K}(\boldsymbol{\phi}_{(2),Q,S},\boldsymbol{\phi}_{(2),Q,S})^{-1}\,\mathfrak{N}_S(\boldsymbol{\phi}_{(2),Q,S},\boldsymbol{\phi}_{(2),R,S})\,\mathbf{K}(\boldsymbol{\phi}_{(2),R,S},\boldsymbol{\phi}_{(2),R,S})^{-1} \\
&\quad \otimes \ldots
\end{aligned} \tag{4.2.2}$$

so that this problematic term reduces to the Kronecker product of the equivalent term over each individual hyperparameter. This means that we only have to perform the Cholesky factorisation and multiplication of matrices of size equal to the number of samples for each hyperparameter. As an example, if we need, say, 100 samples for each of our 20 hyperparameters, we only ever need to perform our expensive $O(\eta^3)$ operations on matrices of size 100, rather than on the full matrix of size 100^{20}. This represents one way of evading the ‘curse of dimensionality’.

In limiting ourselves to sampling only once, we have committed to sampling before the first time step, before we have actually seen any data. As such, we are limited to sampling purely from our hyperparameter priors $p(\,\phi\,|\,I\,)$. This is not unjustifiable - as demonstrated by (3.6.1), these remain as a multiplicative factor in the integrands regardless of how much data is obtained. Hence in ruling out samples where the prior is small, it’s reasonable to hope we have only ignored points where the entire integrand would have been similarly insignificant.

Having considered hyperparameter priors, we now turn to another of the terms in our integrands- the hyperparameter likelihoods $p(\,\boldsymbol{y}_D\,|\,\phi,\,I\,)$. Likelihoods are commonly smaller than machine precision and so for numerical reasons we prefer to instead work with log-likelihoods. Naturally, these will then be exponentiated whenever required. Noting that the determinant of any matrix can be expressed as the product of its eigenvalues, which for a triangular matrix are simply its diagonal elements, we can rewrite (3.1.1) to give the log-likelihood as

$$\begin{aligned}
\log p(\,\boldsymbol{y}_{1:n}\,|\,\phi,\,I\,) &= -\frac{1}{2}\log\det 2\pi\,\mathbf{V}(\boldsymbol{t}_{1:n},\boldsymbol{t}_{1:n}) - \frac{1}{2}\mathfrak{a}_n^{\mathrm{T}}\mathfrak{a}_n \\
&= -\sum_l \log\sqrt{2\pi}\,\mathbf{R}(\boldsymbol{t}_{1:n},\boldsymbol{t}_{1:n})_{l,l} - \frac{1}{2}\mathfrak{a}_n^{\mathrm{T}}\mathfrak{a}_n
\end{aligned} \tag{4.2.3}$$

Hence it can be seen that an update of the log-likelihood of our hyperparameters is also possible

$$\log p(\,\boldsymbol{y}_{1:n}\,|\,\phi,\,I\,) = \log p\big(\,\boldsymbol{y}_{1:n-1}\,\big|\,\phi,\,I\,\big) - \frac{1}{2}\,|\boldsymbol{y}_n|\log 2\pi - \sum_{l=1}^{|\boldsymbol{y}_n|}\log\mathbf{U}_{l,l} - \frac{1}{2}\mathfrak{c}_i^{\mathrm{T}}\mathfrak{c}_i \tag{4.2.4}$$

where $|\boldsymbol{y}_n|$ is the number of new data points obtained.

4.3 Discarding Data

In practice, a GP requires very small amounts of data to produce good estimates. In the context of prediction, often as few as the last several data points will prove necessary to produce reasonable estimates for the predictants. One reason for this can be seen in the very light tails of the commonly used covariance functions displayed in Figure 3.2. Hence only points within a handful of input scales will prove at all relevant to the point of interest.

Further, reducing the size of our data set is desirable for a number of reasons. As mentioned in Section 3.5, a smaller data set produces a better conditioned covariance and can be expected to give rise to smaller associated errors. Secondly, despite our efforts in Section 4.1, the cost of an update remained on the order of $O(n^2)$. While this certainly represents an improvement, given a finite computer, it still imposes a distinctly finite limit on the number of time steps we can possibly consider. Clearly this is unsatisfactory - we'd like our prediction algorithms to be capable of running more or less indefinitely. Finally, our finite computer will also only have finite memory; as such we have a certain limit to the amount of data we can store, regardless of any wishes otherwise. Hence we seek sensible ways of discarding information once it has been rendered 'stale'.

One pre-eminently reasonable measure of the value of data is the uncertainty we still possess after learning it. In particular, we are interested in how uncertain we are about $\boldsymbol{x}_\star$, as given by the covariance of our Gaussian mixture (3.8.19). Our approach is to drop our oldest data points, those which our covariance deems least relevant to the current predictant, until this uncertainty becomes unacceptably large. Note that the uncertainty of a GP drops monotonically with the addition of more data. So, after we've dropped some data, the resultant increase in uncertainty can only ever be ameliorated with time. We do not have to consider the long-term consequences of our discarding of data, only how it affects our uncertainty right now.

Let's see how we can drop old data without having to completely redetermine all the quantities in our model. In effect, we now have do perform a downdate for each hyperparameter sample, the opposite of what we were trying to achieve in Section 4.1. Our new problem is to find $\mathbf{R}(\boldsymbol{t}_{2:n}, \boldsymbol{t}_{2:n})$, given that we know $\mathbf{R}(\boldsymbol{t}_{1:n}, \boldsymbol{t}_{1:n})$. We write

$$\begin{bmatrix} \mathbf{V}(\boldsymbol{t}_1, \boldsymbol{t}_1) & \mathbf{V}(\boldsymbol{t}_1, \boldsymbol{t}_{2:n}) \\ \mathbf{V}(\boldsymbol{t}_{2:n}, \boldsymbol{t}_1) & \mathbf{V}(\boldsymbol{t}_{2:n}, \boldsymbol{t}_{2:n}) \end{bmatrix} = \mathbf{V}(\boldsymbol{t}_{1:n}, \boldsymbol{t}_{1:n}) = \mathbf{R}(\boldsymbol{t}_{1:n}, \boldsymbol{t}_{1:n})^{\mathrm{T}}\, \mathbf{R}(\boldsymbol{t}_{1:n}, \boldsymbol{t}_{1:n}) \tag{4.3.1}$$

where we know the Cholesky factor to be

$$\mathbf{R}(\boldsymbol{t}_{1:n}, \boldsymbol{t}_{1:n}) = \begin{bmatrix} \mathbf{R}(\boldsymbol{t}_1, \boldsymbol{t}_1) & \mathbf{S}' \\ 0 & \mathbf{U}' \end{bmatrix} \tag{4.3.2}$$

where $\mathbf{U}'$ is upper triangular. Hence

$$\mathbf{V}(\boldsymbol{t}_{2:n}, \boldsymbol{t}_{2:n}) = \mathbf{R}(\boldsymbol{t}_{2:n}, \boldsymbol{t}_{2:n})^{\mathrm{T}}\, \mathbf{R}(\boldsymbol{t}_{2:n}, \boldsymbol{t}_{2:n}) = \mathbf{S}'^{\mathrm{T}}\mathbf{S}' + \mathbf{U}'^{\mathrm{T}}\mathbf{U}' \tag{4.3.3}$$

Fortunately, the solution for $\mathbf{R}(\boldsymbol{t}_{2:n}, \boldsymbol{t}_{2:n})$ can be efficiently found by exploiting the special structure of (4.3.3). In particular, there are algorithms to perform exactly this kind of downdate of a Cholesky factor, which we'll denote as $\mathrm{cholupdate}$, after the MATLAB®[The MathWorks, 2007] function. Hence

$$\mathbf{R}(\boldsymbol{t}_{2:n}, \boldsymbol{t}_{2:n}) = \mathrm{cholupdate}(\mathbf{U}',\, \mathbf{S}') \tag{4.3.4}$$

We'll also need to downdate $\mathfrak{a}$, but unfortunately, there is no way to do this more efficient than the direct solution

$$\mathfrak{a}_{2:n} = \mathbf{R}(\boldsymbol{t}_{2:n}, \boldsymbol{t}_{2:n})^{\mathrm{T}} \setminus (\boldsymbol{y}_{2:n} - \mu(\boldsymbol{t}_{2:n})) \tag{4.3.5}$$

Naturally there is no computational benefit here to be gained in downdating our log-likelihood, whose dimension remains constant regardless of the time step. Hence, at no cost, we can retain all knowledge of old data that is pertinent to our marginalisation of hyperparameters[2]. This is important as it means that as we progress through time, we can continue to refine our weights over the hyperparameter samples, improving our model. The somewhat glib statement made earlier to the effect that a GP needs only a few data points is true only if it has sufficient knowledge of the hyperparameters. Fortunately, we can achieve this by simply continuing to update our log-likelihoods.

We have just seen that downdating involves a reasonable amount of effort for a single hyperparameter sample. As such, it may not be feasible to evaluate the full Gaussian mixture uncertainty (3.8.19), involving downdating for all hyperparameter samples, each time we wish to trial the discarding of a datum. A more practical approach is to compute the uncertainty in terms of only one or a few of the higher weighted samples. Once it has been determined how much data is to be dropped, a single downdate can be performed for each hyperparameter sample, a time step advanced and the prediction algorithm continued.

We've referred above to the correlations expressed by a covariance function decreasing with separation in time. Of course, this will not be true for the periodic covariance of a periodic signal - old data will remain pertinent. In such a case, we could consider implementing a more sophisticated search strategy to determine which data points, if not the oldest, are least relevant as measured by our uncertainty. In practice, all physical

[2] It should be noted, however, that after we have dropped a data point, all subsequent updates to the log-likelihood (4.2.4) will be made using covariances computed over the data set without the dropped data point. Of course these covariances should embody only weakly correlations with older data, and so the effects of discarding data should be small.

periodic signals will be modulated by a ‘forgetting’ covariance term, such as a squared exponential. No real periodic signal is so periodic that exceedingly old data is equally as informative as new data.

In conclusion, this proposed method of discarding data will allow us to retain only a manageable data set. At the same time, using a permissible threshold of uncertainty as a criterion ensures that discarding will not have a significant effect on our predictions. This allows a principled way of introducing a ‘windowing’ approach to our data series. Of course, the same methods of downdating can be used for other measures - for example, we may wish to put in place a maximum allowable size on our data set, such that it never exceed available memory. This represents just as valid a utility as that of minimising uncertainty - indeed, we are constantly forced to make decisions with constraints on how long we have available for computation. We can use these methods to ensure that the computational cost of each time step reaches a plateau. Hence we can envisage our GP algorithm continuing indefinitely, producing good predictions within allowable time constraints.

4.4 Active Data Selection

Just as we’ve used our uncertainty to guide our discarding of data, we can consider using it as a guide as to which data to collect in the first place. What we want to achieve is active data selection, whereby the GP is able to decide for itself which observations it should take. Of course, the uncertainty alone is not a sufficient measure in this regard - if our goal was solely to minimise our uncertainty, we’d simply take as many observations as physically possible. In practice, taking observations is associated with some cost, be it the energy required to power a sensor, or, more pertinently, the computational cost associated with additional data. As such, we state our desiderata as the maintenance of our uncertainty below a certain threshold while taking as few observations as possible.

As a potential context for this problem, imagine that we have multiple sensors, each measuring a time-dependent variable. We must choose both when to next take an observation and also which sensor we should draw this observation from. To solve the first part of the problem, we must determine how long we can go without new data before our uncertainty grows (for a GP, it will increase monotonically) beyond our specified permissible threshold. To do this, we evaluate what our uncertainty (3.8.19) is about a certain discrete set of sample future times. This evaluation is a computationally expensive operation - as mentioned earlier, we may wish to evaluate the covariance only in terms of a few of the highest weighted hyperparameter samples. As such, it is impractical to search for the precise time at which the uncertainty reaches the threshold by using simply these evaluations. Instead, we wish to use a computationally cheaper function with which to do our exploration.

Our solution is to place another GP on our uncertainty[3]. After evaluating exactly what this uncertainty is for a set of sample times, we can use those evaluations as data to inform this new GP. The mean of this GP offers a cheap way to estimate what the uncertainty would actually be about any other, non-evaluated time. This mean can then be used in any non-linear optimisation algorithm to approximate when the uncertainty grows above our specified threshold. To assist this optimisation, note that we can also readily evaluate the first and second derivatives of the new GP.

To see this, note that we can use (A.1.5)-(A.1.8) to determine our beliefs about the derivative of a function over which we have a GP. We define the del operator as being the potentially infinite vector of derivative operators with respect to the inputs

$$\boldsymbol{\nabla} \triangleq [\ldots, \frac{\partial}{\partial t_{i-1}}, \frac{\partial}{\partial t_i}, \frac{\partial}{\partial t_{i+1}}, \ldots] \tag{4.4.1}$$

Hence we use $\mathrm{diag}(\boldsymbol{\nabla})$ as we would any other linear transform. We define $\dot{\boldsymbol{x}}$ to be the expected vector of derivatives

$$\dot{\boldsymbol{x}} \triangleq [\ldots, \frac{\partial x(t_{i-1})}{\partial t_{i-1}}, \frac{\partial x(t_i)}{\partial t_i}, \frac{\partial x(t_{i+1})}{\partial t_{i+1}}, \ldots] \tag{4.4.2}$$

Hence, if

$$\begin{aligned} p(\,\boldsymbol{x} \mid I\,) &= \mathbf{N}(\boldsymbol{x};\, \boldsymbol{\mu},\, \mathbf{K}) \\ p(\,\dot{\boldsymbol{x}} \mid \boldsymbol{x}\, I\,) &= \delta(\dot{\boldsymbol{x}} - \mathrm{diag}(\boldsymbol{\nabla})\, \boldsymbol{x}) \end{aligned} \tag{4.4.3}$$

we have

$$p(\,\boldsymbol{x},\, \dot{\boldsymbol{x}} \mid I\,) = \mathbf{N}\left(\begin{bmatrix}\boldsymbol{x}\\ \dot{\boldsymbol{x}}\end{bmatrix};\, \begin{bmatrix}\boldsymbol{\mu}\\ \mathrm{diag}(\boldsymbol{\nabla})\,\boldsymbol{\mu}\end{bmatrix},\, \begin{bmatrix}\boldsymbol{\Lambda} & \boldsymbol{\Lambda}\,\mathrm{diag}(\boldsymbol{\nabla})^{\mathrm{T}}\\ \mathrm{diag}(\boldsymbol{\nabla})\,\boldsymbol{\Lambda} & \mathrm{diag}(\boldsymbol{\nabla})\,\boldsymbol{\Lambda}\,\mathrm{diag}(\boldsymbol{\nabla})^{\mathrm{T}}\end{bmatrix}\right) \tag{4.4.4}$$

Hence, if we place a GP on a function $x(t)$, this distribution is jointly Gaussian with those of all its derivatives $\dot{x}(t)$, $\ddot{x}(t)$ and so on and also those of all its antiderivatives $\int_0^t x(t')\mathrm{d}t'$, $\int_0^t \int_0^{t'} x(t'')\mathrm{d}t''\mathrm{d}t'$ and so on. The full integral we considered in Section 3.7 can be considered a limiting case of this. Naturally, required marginals and conditionals can quickly be derived from (4.4.4).

Given these derivatives, the required observation time can be efficiently determined. The remaining task ahead of us is the choice of which observation to actually take. Our solution here is to imagine taking an observation from each sensor in turn, and then use exactly the procedure above to determine how long it would be until we would then need to take an observation yet again. Our preferred choice is the one that yield the longest such time. This is a greedy approach; we do not consider searching over all possible future schedules of

[3] If you like, our algorithm will model itself, just as we model ourselves when making decisions - ‘will buying this new car really make me happier?’.

observations. Of course, such a strategy is justified by the fact that the covariance will almost always be a well-behaved function of the inputs. It seems unlikely that a selection of observation that proves very informative in the short-term will come to be seen as a poor choice in the long-term.

How is this possible, however, given that we don't actually receive an observation from each of those sensors when we perform this feat of imagination? Well, our solution is feasible only so long as our measure of uncertainty is that of a single highly weighted sample set of hyperparameters. It's reasonable to expect this sample to be representative of how the uncertainty would grow under any of the other significantly weighted samples. Our approach is to take the model, represented by our hyperparameters, as fixed, and investigate only how different schedules of observations affect our predictions within it. Note that the GP for a single set of hyperparameters, given by (3.5.3), has a covariance that is a function only of the times of observation, not on what the observations actually were. Hence our uncertainty can be quickly determined without having to speculate about what data we might possibly collect.

Chapter 5

Weather Sensor Networks

5.1 Tracking

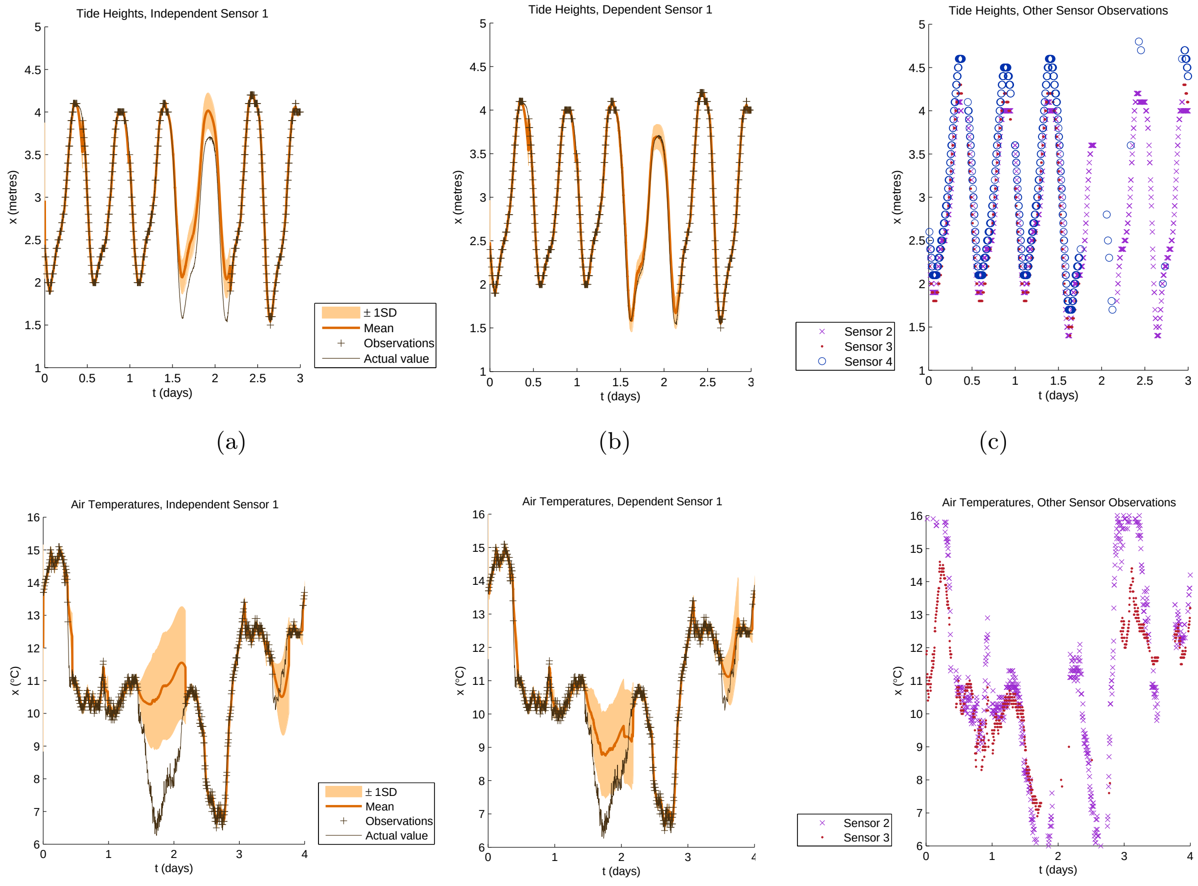
As a testbed for the algorithms proposed above, we now turn to a network of weather sensors located off the coast near Southampton [Chan, 2000]. In particular, we consider readings taken by four sensors located at Bramble Pile, Southampton Dockhead, Chichester Bar and Camber Pile, which we'll label as sensors 1 through 4 respectively. Each sensor is capable of taking readings of several variables associated with local weather conditions, such as wind speed, tide height and air temperature. A new reading for these variables is recorded every minute and stored, while every five minutes the readings are transmitted and uploaded to the internet. From there, we can readily download data for use by our algorithm.

The data we receive from these sensors is subject to many of the problems we discussed earlier. In particular, our four sensors are not separated by more than a few tens of kilometres and hence can be expected to experience reasonably similar weather conditions. As such, their readings are likely to display a strong degree of correlation. However, there may be a slight difference in the times at which each sensor experiences, say, a particular weather front, as it moves along the southern coast of England. Hence a sensor's readings may exhibit some latency relative to another's.

Finally, it is far from uncommon for there to be missing data. This can be the result of several possible faults, but probably the most frequent are a failure of the transmission from the data collection site and/or the resulting upload to the internet. In these cases, of course, it is still possible to later access the archives of the data collection site. Hence we have a perfect way to retrospectively compare our predictions for the 'missing' readings to their actual values.

To demonstrate how our algorithm performs in such occurrences of missing data, refer to Figure 5.1. This illustrates real dropouts of the uploaded data from sensor 1 due to extreme weather conditions. These conditions are, however, exactly what we are most interested in studying. It can be seen that during the most pronounced period of missing data, a strong wind from the North has actually depressed the tide by a few hundred millimetres.

Of course, weather variables such as tide heights are generally well-predicted by models built upon historical data, giving rise to published tide tables. However, these tables do not account for unusual, local conditions that we can detect from our sensors. Such disturbances can be significant, as shown by the data set in 5.1. Moreover, our GP gives a full probabilistic model and as such we can produce error bars on our predictions, valuable information to users of the weather sensor network.



(a) (b) (c)

(d) (e) (f)

Figure 5.1: Weather sensor network, with a notable sensor 1 dropout at $t = 1.45$ days. (a) depicts the GP for tide heights at Sensor 1 assuming all sensors are independent, (b) allows for the possibility of correlations amongst the sensors. For both, the observations available to the GP are depicted along with the actual known values. (c) represents readings from the other sensors available to the GP, demonstrating the strong correlations within the network. (d) - (f) present the equivalent plots for air temperature readings.

Figure 5.1 illustrates the efficacy of our GP prediction in this scenario. For the tides, we used a covariance function that was the sum of a periodic term K_1 and a disturbance term K_2 as

$$K\big([n,\,t],\,[n',\,t']\big) = \mathbf{C}(n,\,n')\,\Big(K_{1,\,h_1=1,\,w_1}\big(t-\boldsymbol{d}(n),\,t'-\boldsymbol{d}(n')\big)+K_{2,\,h_2,\,w_2}\big(t-\boldsymbol{d}(n),\,t'-\boldsymbol{d}(n')\big)\Big) \quad (5.1.1)$$

This form is a consequence of our expectation that the tides would be well modelled by the superposition of a simple periodic signal and an occasional disturbance signal due to exceptional conditions.

Using the marginalisation techniques presented above, we can determine the weights (3.8.16) associated with predictions made using a trial set of different values for each hyperparameter. It is commonly the case that after a significant volume of data, the majority of the weight will be concentrated at a single trial hyperparameter - the posterior distribution for the hyperparameter $p(\,\phi\,|\,\boldsymbol{y}_D,\,I\,)$ will be close to a delta function at that value. In this case, we say informally that we have *learned* the hyperparameter to have that single value.

In this sense, the period w_1 of the first term K_1 was unsurprisingly learnt as being about half a day, whereas for the disturbance term the time scale w_2 was found to be two and a half hours. Note that this latter result is concordant with our expectations for the time scales of the weather events we intend our disturbance term to model. Both covariance terms were of the Matérn class (3.2.2) with $\nu = \frac{5}{2}$. Of course, for a better fit over the course of, say, a year, it would be simple to additionally incorporate longer-term drifts and periods.

In (5.1.1), n and $n' = 1,\,\ldots,\,4$ refer to the labels of the sensors that have taken the readings at t and t' respectively. These labels represent simply an additional input dimension. To cope with this multi-dimensional input space, we have chosen a covariance (5.1.1) that is the Hadamard product of a covariance function over time alone and a covariance function over sensor label alone; this product we know from Section 3.3 to be a valid covariance itself. We leave our covariance over sensors in the most naive and general form by writing

$$\mathbf{C} = \mathrm{diag}(\boldsymbol{l})\,\mathbf{s}^{\mathrm{T}}\,\mathbf{s}\;\mathrm{diag}(\boldsymbol{l}) \quad (5.1.2)$$

where $\mathbf{s}$ is the correlation matrix of the spherical parameterisation (3.4.2). Unsurprisingly, our sensors were determined to be very strongly correlated, with $\mathbf{s}^{\mathrm{T}}\,\mathbf{s}$ containg elements all very close to one. $\boldsymbol{l}$ gives an appropriate length scale for each sensor. Over this data set, sensor 3 was found to have a length scale of 1.4m, with the remainder possessing scales of close to 1m. Note that h_1 and h_2 in (5.1.1) are dimensionless ratios, serving as weightings between our various covariance terms. Hence we can set $h_1 = 1$ without loss of generality, upon which we find h_2 to be around 0.2. Hence weather events were observed to have induced changes in tide height on the order of 20%.

We also make allowances for the prospect of relative latency amongst the sensors by incorporating the delay variable $\boldsymbol{d}$. With an entry for each sensor, $\boldsymbol{d}$ gives a fixed relative delay associated with each sensor

- clearly only 3 variables are needed to fully describe $\boldsymbol{d}$. This delay variable tells the GP that, for example, a measurement from sensor 1 at time t is most strongly correlated with the reading from sensor 2 at time $t - \boldsymbol{d}(2) + \boldsymbol{d}(1)$. We found that the tide signals at sensors 3 and 4 were delayed by about 50 minutes relative to both sensors 1 and 2. This makes physical sense - sensors 1 and 2 are quite exposed to the ocean, while water has to run further through reasonably narrow channels before it reaches sensors 3 and 4. For other systems, these delays could also be considered to be functions of time - perhaps sometimes one sensor would lag relative to another, where other times it may lead. This would be the case if we were measuring a variable such as rainfall, with weather fronts approaching from different directions and different speeds.

Clearly air temperature represents a more complicated data set. Note that sensor 4 does not record air temperature, and is hence discluded from consideration here. Here, we used a covariance function of the form

$$K\big([n,\,t],\,[n',\,t']\big) = \mathbf{C}(n,\,n')\,\Big(K_{1,\,h_1=1,\,w_1}\big(t-\boldsymbol{d}(n),\,t'-\boldsymbol{d}(n')\big) + K_{2,\,h_2,\,w_2}\big(t-\boldsymbol{d}(n),\,t'-\boldsymbol{d}(n')\big)\Big) \\ + \delta\big(n-n'\big)\;K_{3,\,h_3,\,w_3}\big(t-\boldsymbol{d}(n),\,t'-\boldsymbol{d}(n')\big) \quad (5.1.3)$$

here K_1 is a periodic term, whose period is found to be almost exactly 1 day, describing nightly dips in temperature. K_2 describes longer-term drifts in temperature, and was determined to have a time scale of 3 days, and a magnitude h_2 just over twice that of the periodic term. Finally, K_3 represents much higher frequency disturbances, being found to possess a time scale of 4 hours. Note that this final term is not correlated amongst sensors as were the other two: these disturbances appear to be local to each sensor. The correlations that are indicated by (5.1.3) were again found to be very strong, and the length scales $\boldsymbol{l}$ of the covariance matrix were found to be 2°C, 6°C and 2.6°C respectively. These give the scales of the periodic fluctuations and longer-term drifts in the air temperature at each sensor. For comparison, the scale of the disturbances h_3 were found to be about 1°C. Again, all covariance functions were of the the Matérn class, with ν equal to $\frac{5}{2}$, $\frac{5}{2}$ and $\frac{3}{2}$ respectively for the three terms.

We can now return to discussion of the results depicted in Figure 5.1. At time t, this figure depicts the posterior distribution of the GP about the weather at time t, conditioned on the knowledge of all observations prior to t - the plot depicts a dynamic tracking process. Note there are brief intervals of missing data for Sensor 1 just after both of the first two peak tides. During the second interval, the GP's predictions for the tide are notably better than for the first - the greater quantity of data it has observed by the time of the second interval allow it to produce more accurate predictions. With time, the GP is able to build successively better models for the series.

Note that the hyperparameters mentioned above were all learned by the GP prior to the depicted data set, with the single exception of the correlations amongst sensors. For the results in Figures 5.1a and 5.1d, the GP was forced to assume that all sensors were independent: the correlation matrix s was the identity. For Figures 5.1b and 5.1e, the angles θ that determine the correlation matrix were given a vague prior and then marginalised at each of the time steps displayed.

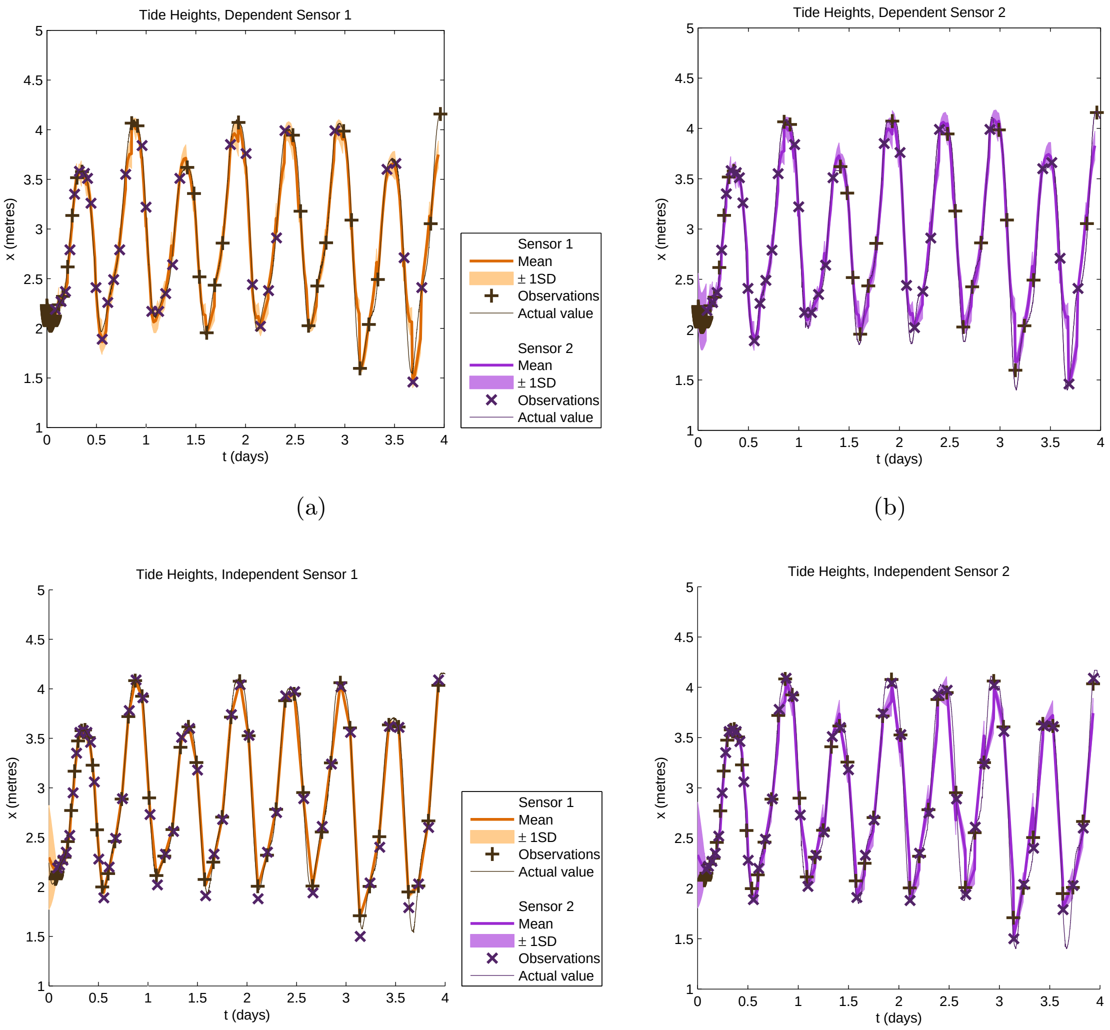
We can now compare the performance of the GP given these independent and dependent assumptions. When sensor 1 drops out at $t = 1.45$ days, note that the GP that assumes sensor 1 is independent quite reasonably assumes that the tide and air temperature will continue to do more or less what it has seen before. However, the GP can achieve better results if it is allowed to benefit from the knowledge of the other sensor's readings during this interval of missing data from sensor 1. In the dependent case, by $t = 1.45$ days, the GP has successfully determined that the sensors are all very strongly correlated. Hence, when it sees a disturbance in sensors 2, 3 and 4, the correlations it knows exist in this term allow it to induce a similar disturbance to the readings at sensor 1. This allows it to produce the significantly more accurate predictions of Figures 5.1b and 5.1e for the missing data interval, with associated smaller error bars. The conclusion is clear: our GP is able to benefit from as much information as we can supply it.

Note that in the results above, we have not considered correlations between variables, such as between the readings of tide height and air temperature. There is, however, absolutely no theoretical obstacle that would prevent us doing so - the methods above would apply equally to this case. This would further improve the performance of our predictions, allowing the GP to use knowledge of an abnormally strong wind to reduce predicted peak tides.

5.2 Active Data Selection

We now turn to a demonstration of our active data selection algorithm. Using the fine-grained recorded data, we can simulate how our GP would have chosen its observations had it been in control of the sensors. With readings taken by each sensor every minute, we simply round the observation times requested by the GP down to the nearest minute. This conservatism should ensure that our specified uncertainty threshold is not exceeded. Results from the active selection of observations from sensor 1 and 2, only, are displayed in Figure 5.2. Again, these plots depict dynamic choices: at time t, the GP must decide when next to observe, and from which sensor, given knowledge only of the observations recorded prior to t.

Consider first the dependent case, Figures 5.2a and 5.2b, in which the GP is allowed to contemplate correlations between the two sensors. As before, all hyperparameters are assumed known other than these correlations. Note that a large number of observations are taken initially as the dynamics and correlations



(a)

(b)

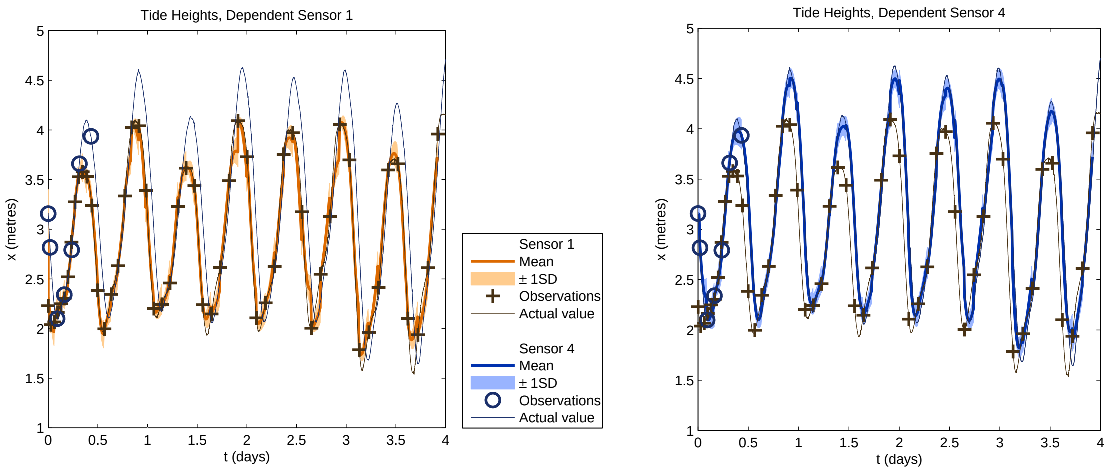
(c)

(d)

Figure 5.2: Weather sensor network with observations actively selected to maintain the standard deviation below 0.15m. (a) depicts the GP for tide heights at sensor 1, (b) at sensor 2, allowing for the possibility that the readings from the sensors may be correlated. (c) and (d) represent equivalent plots where sensors are assumed independent.

are learnt; later, a low but constant rate of observation is chosen. Observations are roughly evenly distributed between sensors. Peaks and troughs are adjudged to be of particular importance - it is at these locations that our uncertainty would otherwise be greatest, as shown by Figure 5.1. For the independent case, Figures 5.2c and 5.2d, the GP is forced to make similar observations from both sensors. Consequently 127 observations are required to keep the uncertainty below the specified tolerance, where only 66 were required in the dependent case depicted in (a) and (b). This represents another clear demonstration of how our prediction is able to benefit from the readings of multiple sensors.

We also consider the selection of observations from sensors 1 and 4, as illustrated in Figure 5.3. This



(a)

(b)

Figure 5.3: Weather sensor network with observations actively selected to maintain the standard deviation below 0.15m. (a) depicts the GP for tide heights at sensor 1, (b) at sensor 4, allowing for the possibility that the readings from the sensors may be correlated.

data set is notable for the tide height at sensor 4, Figure 5.3b, being slightly delayed relative to sensor 1, Figure 5.3a. Here the GP is forced to marginalise over both the correlations between sensors and any latency that may exist between them. Note that after an initial learning phase as the dynamics, correlations, and delays are inferred, the GP chooses only to sample from the undelayed sensor 1. As sensor 2 is found to be strongly correlated with but delayed by 50 minutes relative to sensor 1, clearly readings from sensor 1 represent the more informative choice. Despite no observations at all being made of the tide at sensor 4, the resulting predictions remain remarkably accurate.

5.3 Pigeon Navigation

The remarkable capacity of pigeons to fly home, even from never before visited locations, remains an active area of biological research [Roberts et al., 2004]. As such, we consider a data set, illustrated in Figure 5.5a, in which multiple pigeons were released multiple times from various locations and their subsequent flight paths back to a common roost tracked by miniature GPS devices. Fuller details of the experiments are available in Roberts et al. [2004].

Figure 5.4 depicts the Bayesian network for this particular data set. We assume that associated with each pigeon g is an idiosyncratic habitual path [Meade et al., 2005], $[\boldsymbol{h}_g, \boldsymbol{h}'_g]$. Here, $\boldsymbol{h}_g$ is the vector of the first (e.g. longitude) dimension of our two dimensional path, $\boldsymbol{h}'_g$ the second (e.g. latitude). These form the path it will try and recreate whenever re-released in the same area. However, due to a multitude of dynamic environmental variables - the wind, the time of the season, new scents and so on - it may not be able to exactly retrace $[\boldsymbol{h}_g,\, \boldsymbol{h}'_g]$ for each new iteration p and instead follows the path $[\boldsymbol{x}_{p,g},\, \boldsymbol{x}'_{p,g}]$, where again we have separated the first and second dimensions of the path. Perhaps unsurprisingly, we choose to model the functions of time $h_g(t)$, $h'_g(t)$, $x_{p,g}(t)$ and $x'_{p,g}(t)$ as GPs, as in Figure 5.5b.

Associated with both, then is an input scale - a time scale - σ. Note the slight change in notation from what we have previously have referred to as w. This controls the smoothness of the paths and hence is determined by the fundamental physics of pigeon flight. In particular, σ can be thought of as a measure of the time it takes a pigeon to turn through a set angle once in flight. For large σ, the pigeon is slow to turn; hence the paths will be smooth. Similarly, λ_H and λ_X represent length scales for the learnt and iterated paths respectively. Note that we assume the paths are isotropic, meaning that the length scales in both output dimensions are taken as identical. These scales again determined by the physics of flight; they are related to how far we expect the pigeon to move in a given time interval. More precisely, they represent the scales of variation of the pigeon around our expected mean path. For λ_X, this mean is the learned path - a large λ_X means that we expect pigeons to vary wildly and quickly from what they have learned. For λ_H, the mean will simply be a straight line between the pigeon's release point and its roost $\boldsymbol{\mu}$. If we possess large λ_H, then, we expect the pigeons to rapidly diverge from that simple path. We group these three hyperparameters into the single variable ϕ.

Given ϕ, we assume for simplicity that the two dimensions of the paths are independent functions of time. Hence we will write the distributions for $h_g(t)$ and $x_{p,g}(t)$ alone - given the isotropicity of the paths, the distributions for the other dimension will be identical, and the joints will be the simple product of the two dimensions. Note that our distributions will be conditioned on the fact that we know each path will begin at the release point and finish at the roost - we incorporate this knowledge into I. This information effectively forms two data points, constraining the form of our GP. As such, we can use the usual GP regression formulae (3.5.1)

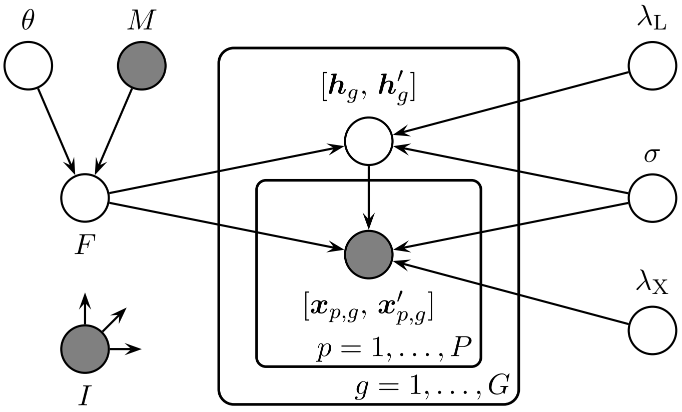
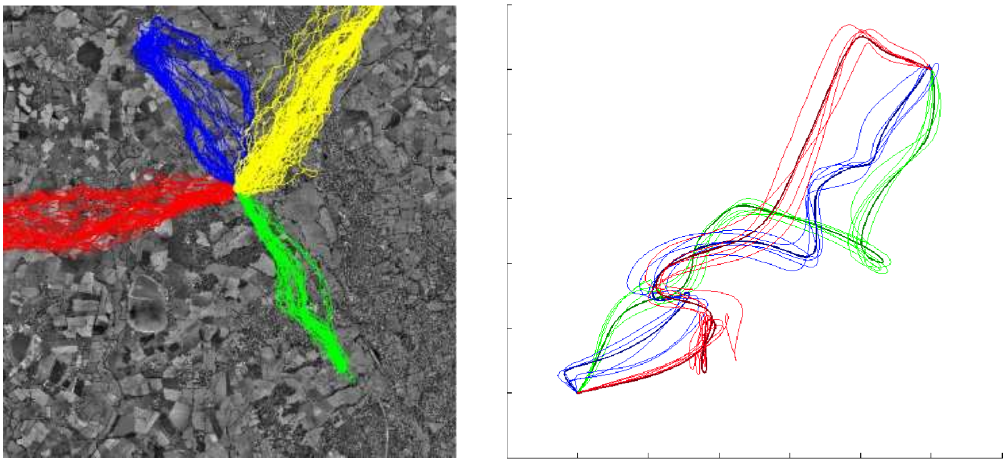

θ
M
λ_L
$[\boldsymbol{h}_g,\, \boldsymbol{h}'_g]$
σ
F
λ_X
$[\boldsymbol{x}_{p,g},\, \boldsymbol{x}'_{p,g}]$
$p = 1, \ldots, P$
$g = 1, \ldots, G$
I

Figure 5.4: Bayesian network for pigeon navigation.



(a) (b)

Figure 5.5: (a) Recorded paths of 29 pigeons, released from 4 separate locations between 3 and 12 times each. (b) pseudo-random draws from a GP over the paths for three different pigeons - the dark lines represent habitual paths $[\boldsymbol{h}_g,\, \boldsymbol{h}'_g]$, the lighter lines several paths $[\boldsymbol{x}_{p,g},\, \boldsymbol{x}'_{p,g}]$.

to give

$$
\begin{aligned}
p(\, \boldsymbol{h}_g \,|\, \sigma,\, \lambda_H,\, I \,) &\triangleq \mathbf{N}\big(\boldsymbol{h}_g;\, \boldsymbol{\mu},\, \lambda_H^2\, \mathbf{C}_\sigma\big) \\
p(\, \boldsymbol{x}_{p,g} \,|\, \boldsymbol{h}_g,\, \sigma,\, \lambda_X,\, I \,) &\triangleq \mathbf{N}\big(\boldsymbol{x}_{p,g};\, \boldsymbol{h}_g,\, \lambda_X^2\, \mathbf{C}_\sigma\big)
\end{aligned}
\tag{5.3.1}
$$

We define the time at which the pigeon is released as $t = 0$ and the time at which it reaches the roost as $T = 0$. Note that, of course, T will be different for each path. However, we can simplify our calculations dramatically by assuming that T is equal for all paths connecting the same release point and roost. This allows us to define the single covariance

$$
\mathbf{C}_\sigma(\boldsymbol{t}_\star, \boldsymbol{t}_\star) \triangleq \mathbf{K}_\sigma(\boldsymbol{t}_\star, \boldsymbol{t}_\star) - \mathbf{K}_\sigma\left(\boldsymbol{t}_\star, \begin{bmatrix}0\\T\end{bmatrix}\right) \mathbf{K}_\sigma\left(\begin{bmatrix}0\\T\end{bmatrix}, \begin{bmatrix}0\\T\end{bmatrix}\right)^{-1} \mathbf{K}_\sigma\left(\begin{bmatrix}0\\T\end{bmatrix}, \boldsymbol{t}_\star\right) \tag{5.3.2}
$$

and $K_\sigma(\cdot,\cdot)$ is an appropriate stationary covariance function with an output scale equal to one and with an input scale of σ. Note the appropriate length scales λ can then be simply expressed as a multiplicative constant to this covariance matrix.

The remaining nodes in Figure 5.4 relate to other factors we believe may influence the pigeon's navigation. Namely, M is the true map of the region the pigeons fly over, which of course we know. Hence for notational convenience, M will be incorporated into I henceforth. However, the map actually perceived by the pigeons F can be expected to be quite different. For example, pigeons possess some limited visual range and hence may only be able to use landmarks for navigation if they are within some particular radius. Such parameters form the set θ and ultimately form the object of our interest. For now, however, we limit our inference to determining a posterior for the GP hyperparameters ϕ given our data set of paths $\big[[\boldsymbol{x}_{p,g},\, \boldsymbol{x}'_{p,g}];\, \forall p,\, g\big]$.

Once determined, these can be used to generate pseudo-random draws from our GPs, to simulate pigeon paths. Assuming the map itself does not greatly influence the GP hyperparameters, such simulated paths will of course be uninfluenced by any particular details of the map. Such paths can then be compared to the actual determined pigeon paths, as a guide for biologists interested in the influence of various map features.

Now consider

$$\begin{aligned} p(\,[\boldsymbol{x}_{p,g},\,\forall p]\,|\,\phi,\,I\,) &= \int \mathrm{d}\boldsymbol{h}_g\, p(\,\boldsymbol{h}_g\,|\,\sigma,\,\lambda_H,\,I\,)\prod_{p=1}^{P} p(\,\boldsymbol{x}_{p,g}\,|\,\boldsymbol{h}_g,\,\sigma,\,\lambda_X,\,I\,) \\ &= \int \mathrm{d}\boldsymbol{h}_g\, \mathbf{N}(\boldsymbol{h}_g;\,\boldsymbol{\mu},\,\lambda_H^2\,\mathbf{C}_\sigma)\,\mathbf{N}\Big([\boldsymbol{x}_{1,g},\,\ldots,\,\boldsymbol{x}_{P,g}]^{\mathrm{T}}\,;\,\mathbf{1}_{P,1}\otimes\boldsymbol{h}_g,\,\lambda_X^2\,\mathbf{I}_P\otimes\mathbf{C}_\sigma\Big) \end{aligned} \tag{5.3.3}$$

and similarly for the other dimension. We write the dimension of $\boldsymbol{x}$ as Q, where $Q = 2T$ for two-dimensional paths of duration T. Hence, using the mixed product property of the Kronecker product (Appendix A.2) allows us to treat the Kronecker product in the mean of the second Gaussian in (5.3.3) as a regular matrix product, thus allowing the use of (A.1.7) to resolve the integral.

$$\begin{aligned} &p(\,[\boldsymbol{x}_{p,g},\,\forall p]\,|\,\phi,\,I\,) \\ &\quad = \mathbf{N}\Big([\boldsymbol{x}_{1,g},\,\ldots,\,\boldsymbol{x}_{P,g}]\,;\,(\mathbf{1}_{P,1}\otimes\mathbf{I}_Q)\,\boldsymbol{\mu},\,\lambda_X^2(\mathbf{I}_P\otimes\mathbf{C}_\sigma)+\lambda_H^2(\mathbf{1}_{P,1}\otimes\mathbf{I}_Q)\mathbf{C}_\sigma(\mathbf{1}_{P,1}\otimes\mathbf{I}_Q)^{\mathrm{T}}\Big) \end{aligned} \tag{5.3.4}$$

We now define the covariance of (5.3.4) as $\boldsymbol{\Gamma}$. Hence with the further use of the mixed product property, we can then write

$$\boldsymbol{\Gamma}_\sigma = \big(\lambda_X^2\mathbf{I}_P + \lambda_H^2\mathbf{1}_P\big)\otimes\mathbf{C}_\sigma \tag{5.3.5}$$

and, using the inversion property of the Kronecker product, the precision matrix as:

$$\begin{aligned} \boldsymbol{\Gamma}_\sigma^{-1} &= \big(\lambda_X^2\mathbf{I}_P + \lambda_H^2\mathbf{1}_P\big)^{-1}\otimes\mathbf{C}_\sigma^{-1} \\ &= \frac{1}{\lambda_X^2}\left(\mathbf{I}_P - \frac{1}{(\frac{\lambda_X}{\lambda_H})^2 + P}\,\mathbf{1}_P\right)\otimes\mathbf{C}_\sigma^{-1} \end{aligned} \tag{5.3.6}$$

(5.3.6)[1] allows us to efficiently evaluate (5.3.4), although of course we work with the cholesky decomposition of $\mathbf{C}_\sigma$ rather than its direct inverse. This has allowed us to take the decomposition of a matrix of dimension only T, rather than the full matrix of dimension PT. Moreover, this covariance is identical for all pigeons released from the the same sites, that is, for all paths of length T. Hence we can reuse our cholesky factor for all g. This permits the analysis of even large data sets.

Of course, it may not be reasonable to assume that our data is noise-free, as has the analysis above. As we've seen before, to incorporate gaussian noise merely implies adding a noise term to the diagonals of

[1] Thanks to Rich Mann for this lemma.

our covariance matrix. Let's define the new, noisy covariance of (5.3.4) as $\mathbf{V}_\sigma \triangleq \mathbf{\Gamma}_\sigma + \epsilon^2\,\mathbf{I}_{PT}$, where ϵ is the standard deviation of the noie. Unfortunately, it can be seen that this immediately ruins our exact inversion (5.3.6). However, so long as ϵ is small relative to the entries of $\mathbf{\Gamma}$, we can employ a Taylor series approximation

$$\mathbf{V}_\sigma^{-1} \approx \mathbf{\Gamma}_\sigma^{-1} - \epsilon^2\,\mathbf{\Gamma}_\sigma^{-2} + \epsilon^4\,\mathbf{\Gamma}_\sigma^{-3} + \ldots \tag{5.3.7}$$

Similarly, the determinant of $\mathbf{V}_\sigma$ can be approximated by using (3.8.5). This allows us to retain our efficient evaluation of (5.3.4) even in the noisy case.

Recall that the quantity of our immediate interest is the posterior $p(\,\phi\,|\,\big[[\boldsymbol{x}_{p,g},\,\boldsymbol{x}'_{p,g}];\,\forall p,\,\forall g\big],\,I\,)$. Forced to summarise this distribution, we elect to compute simply its mean

$$\psi = \int \mathrm{d}\phi_\star\,\phi_\star\,p(\,\phi_\star\,|\,\big[[\boldsymbol{x}_{p,g},\,\boldsymbol{x}'_{p,g}];\,\forall p,\,\forall g\big],\,I\,) \tag{5.3.8}$$

$$= \frac{\int \mathrm{d}\phi_\star\,\phi_\star\,p(\,\phi_\star\,|\,I\,)\prod_{g=1}^{G} p(\,[\boldsymbol{x}_{p,g};\,\forall p]\,|\,\phi_\star,\,I\,)\;p(\,[\boldsymbol{x}'_{p,g};\,\forall p]\,\big|\,\phi_\star,\,I\,)}{\int \mathrm{d}\phi_\star\,p(\,\phi_\star\,|\,I\,)\prod_{g=1}^{G} p(\,[\boldsymbol{x}_{p,g};\,\forall p]\,|\,\phi_\star,\,I\,)\;p(\,[\boldsymbol{x}'_{p,g};\,\forall p]\,\big|\,\phi_\star,\,I\,)} \tag{5.3.9}$$

Unfortunately, the integrals in (5.3.8) are not solvable analytically. As such, we turn again to BMC to help us approximate their values. Note that the form of our problem is identical to that tackled in (the last line of) (3.8.4), with $m_Q(\phi)$ replaced by simply ϕ, and where the expensive function we are unable to evaluate for all ϕ is

$$r(\phi_\star) \triangleq \prod_{g=1}^{G} p(\,[\boldsymbol{x}_{p,g},\,\forall p]\,|\,\phi_\star,\,I\,)\;p(\,[\boldsymbol{x}'_{p,g},\,\forall p]\,\big|\,\phi_\star,\,I\,) \tag{5.3.10}$$

Using (5.3.4) and (5.3.6) or (5.3.7), we can evaluate $\boldsymbol{r}_S \triangleq r(\boldsymbol{\phi}_{S,R})$ for a set of sample hyperparameters $\boldsymbol{\phi}_{S,R}$, and use those samples in order to evaluate our integral. Note that, given that we want to evaluate a large number of hyperparameter samples, the simplification of the covariance we performed above becomes absolutely essential. As before, we use a gaussian prior (3.7.8) and squared exponential covariance (3.7.9). As such, noting that

$$\int m_R(\phi_\star)\,p(\,\phi_\star\,|\,I\,)\;\mathrm{d}\phi_\star = \mathfrak{n}_{S,R}^{\mathrm{T}}\,\mathbf{K}(\boldsymbol{\phi}_{S,R},\boldsymbol{\phi}_{S,R})^{-1}\,\boldsymbol{r}_S \tag{5.3.11}$$

$$\int \phi_\star\,m_R(\phi_\star)\,p(\,\phi_\star\,|\,I\,)\,\mathrm{d}\phi_\star = \mathfrak{m}_{S,R}^{\mathrm{T}}\,\mathbf{K}(\boldsymbol{\phi}_{S,R},\boldsymbol{\phi}_{S,R})^{-1}\,\boldsymbol{r}_S \tag{5.3.12}$$

where, $\forall j \in S_R$

$$\mathfrak{n}_{S,R}(j) \triangleq h_R^2\,\mathbf{N}\big(\phi_j;\,\boldsymbol{\nu},\,\lambda^{\mathrm{T}}\lambda + \mathbf{w}_R^{\mathrm{T}}\mathbf{w}_R\big) \tag{5.3.13}$$

$$\mathfrak{m}_{S,R}(j) \triangleq h_R^2 \int \phi_\star\,\mathbf{N}\big(\phi_j;\,\phi_\star,\,\mathbf{w}_R^{\mathrm{T}}\mathbf{w}_R\big)\;p(\,\phi_\star\,|\,I\,)\,\mathrm{d}\phi_\star \tag{5.3.14}$$

Note that $\mathbf{N}\big(\phi_i;\, \phi_\star,\, \boldsymbol{w}^{\mathrm{T}}\, \boldsymbol{w}\big)$ can be viewed as a valid probability distribution $p(\,\phi_i \mid \phi_\star,\, I\,)$. Hence we can use the product rule to rewrite as

$$\mathfrak{m}_{S,R}(j) = h_R^2\, p(\,\phi_j \mid I\,) \int \phi_\star\, p(\,\phi_\star \mid \phi_j,\, I\,)\, \mathrm{d}\phi_\star \tag{5.3.15}$$

and so, using (A.1.4), we find

$$\mathfrak{m}_{S,R}(j) = h_R^2\, \mathbf{N}\big(\phi_j;\, \boldsymbol{\nu},\, \mathbf{w}_R^{\mathrm{T}}\mathbf{w}_R + \lambda^{\mathrm{T}}\lambda\big)\, \Big(\lambda^{\mathrm{T}}\lambda\, \big(\mathbf{w}_R^{\mathrm{T}}\mathbf{w}_R + \lambda^{\mathrm{T}}\lambda\big)^{-1}\, (\phi_j - \boldsymbol{\nu}) + \boldsymbol{\nu}\Big) \tag{5.3.16}$$

and so, to a zeroth order approximation

$$m_\Psi \approx \frac{\mathfrak{m}_{S,R}^{\mathrm{T}}\, \mathbf{K}(\boldsymbol{\phi}_{S,R}, \boldsymbol{\phi}_{S,R})^{-1}\, \boldsymbol{r}_S}{\mathfrak{n}_{S,R}^{\mathrm{T}}\, \mathbf{K}(\boldsymbol{\phi}_{S,R}, \boldsymbol{\phi}_{S,R})^{-1}\, \boldsymbol{r}_S} \tag{5.3.17}$$

Chapter 6

Conclusions and Future Work

6.1 Conclusions

We have demonstrated how GPs can be used to give a complete probabilistic framework for prediction and tracking. BMC has been used as a principled way of marginalising out our hyperparameters, in an interesting illustration of the sometimes misinterpreted concept of a second-order probability. While GPs are mostly used with fixed data sets, to produce predictions about a fixed set of predictants, we've introduced an iterative formulation that has allowed us to efficiently produce predictions about ever-changing predictants given dynamic time series data.

GPs have allowed us to deal with a number of seemingly problematic features of time series, such as missing, delayed and correlated data. While we explicitly tackled fixed delays between sensors, note also that it matters not at all if we receive data 'late' or out of sequence. So long as data is time-stamped[1] - if we know the associated inputs - it can be trivially incorporated into our predictions, regardless of when it is received. Similarly, a GP is entirely unaffected if our observations are obtained at irregular intervals. More generally, so long as we simply tell the GP what we know about our data, it can always infer the relevant correlations and update its predictions. This is a particular advantage of our approach.

Sensor failure is an unfortunate constant across a wide variety of applications, but presents no difficulties for a GP. Indeed, our results are helped by the GP's ability to learn the correlations between multiple data streams, to readily perform what is sometimes called data fusion [Durrant-Whyte, 2001]. The GP allows us to effectively integrate information from all the sensors in our system in order to compensate for the intermittent failure of some of them. As noted above, it is equally possible to integrate information from different sources - wind and tide, for example. All that is required is to learn an appropriate covariance between them.

Note that the multi-sensor approach taken in 5.1 was deliberately naive. We gave the GP almost no

[1] If data is not time-stamped, then we would need to marginalise out the uncertain ages of those measurements, which form new hyperparameters of our model.

information concerning how our four sensors were to be correlated, forcing it to infer the parameters of the completely general spherical parameterisation. This is desirable for many multi-sensor networks, in which we really are highly ignorant of how their readings are to be related. Unfortunately, this approach scales poorly with the number of sensors in our network - for N sensors we require $\frac{1}{2}N\,(N+1)$ hyperparameters, which must be expensively marginalised.

Inferring correlations between a large number of sensors is feasible only if we can supply some restrictive prior knowledge. For example, we could inform the GP that certain subsets of the sensors are independent, given their large separation, setting to zero certain of our hyperparameters. Another approach would be along the lines of (3.4.3), in which we specify the positions of sensors as simply another input variable. We can then specify a covariance function over the spatial separation of sensors. This will likely have far fewer hyperparameters than the completely general specification of the spherical parameterisation.

We have demonstrated some of the enormous flexibility of our GP approach. Much of the utility of GPs is found in allowing a fit to data for which we do not otherwise possess good models. However, it needs to be stressed that if we do have more informative prior knowledge, particularly if we have a strong physical model for the process, a GP may not reflect the most appropriate choice. While we can incorporate significant prior information into our covariance function, the Gaussian distribution it must ultimately be built into may be an inappropriate choice.

For example, the mean estimate of a GP for a predictant will always be a linear combination of observed predictants, as per (3.5.1). In this estimate, a GP can be thought of as an auto-regressive model; a GP can always be found to match any particular auto-regressive approach. However, it's quite possible for us to possess beliefs about a system such that our mean estimate for a predictant is some non-linear combination of the predictors, such as for a time series we know to be generated by $x(t) = x(t-1)^2$. Our algorithm could still be made to track such a series reasonably well, but its performance would likely be poorer than another algorithm incorporating our true prior knowledge. A GP can, however, always be used to illuminate trends and patterns in new data that can then be used to fit such better physical models. In summary, our approach can be made astoundingly flexible, but it is certainly not a one-stop shop for time series prediction.

However, ours is not the only approach to using GPs for time series prediction. Both Girard et al. [2003] and Wang et al. [2006] use GPs not to model the process as a function of time, as we have, but to model the *transition model* of a Markovian process. That is, they assume each output $x(t)$ is a function of a set number of previous outputs $[x(t-1),\, x(t-2),\, \ldots,\, x(t-L)]$, and perform inference about that function. Such an approach allows the authors superior performance on the kind of non-linear combination problems we have just described. However, this comes at a cost - missing data becomes a much more significant problem, where each function value not observed effectively becoming a new hyperparameter to be marginalised. Similarly, if a function value is observed that does not fall into the exact sequence supposed - say $x(t - \frac{1}{2})$ - it's unclear how it should be incorporated. Similarly for data received late, or from other sensors. In summary, such an approach is interesting, but seemingly fairly brittle and largely unsuitable for the data sets we consider.

6.2 Outline of Future Work

The principled framework that GPs provide lends itself to many extensions. Perhaps the clearest and most desirable extension of the work above is to relax the constraint of a fixed sample set of hyperparameters. Instead, a straitforward extension is to employ *active sampling* of hyperparameters. This could be implemented in a very similar way to that in which we have actively selected our observations - we would use our uncertainty as a guide as to which hyperparameter sample to consider next. In this, we would use the uncertainty in our integration, similar to (3.7.13), as a measure. As mentioned earlier, these higher moments of the distribution for the integral are *only* of relevance when we consider how our beliefs would be affected by considering new samples.

We had previously justified the use of a fixed set of samples on the ground of practicality. By this we meant that whenever we introduce a new sample we must perform the expensive Cholesky decomposition of a data covariance matrix from scratch, without the benefit of the updating rules we have previously described. However, if we are discarding data in the manner described above, this data covariance matrix can be held to a manageable size. Further, we could restrict ourselves to discarding and then replacing only a small number of hyperparameter samples at each time step. This would render active hyperparameter sampling both attractive and practical.

Another ready extension would be to consider fault detection. Given that the GP provides a complete probabilistic framework, it seems possible that we could use this to perform comparison between a model that assumes a correctly functioning sensor and other, faulty models. If we supply the respective costs of incorrectly deciding a sensor is faulty when it is not and deciding a sensor is not faulty when it is, our algorithm can take appropriate action by maximising its utility. One possible problem is that we would really have to perform model comparison between a very large and growing number of models - one in which the sensor was faulty for just its first observation, one in which it was faulty for just its second observation, one in which it was faulty for both and so on. This may cause analytical difficulties. Nonetheless, fault detection and condition monitoring provide a rich vein of applications and we hope that a principled GP approach would prove valuable in those fields.

We now move to some more speculative possible directions. When in BMC we integrate over the space of all possible functions, we are implicitly employing the fruits of functional analysis [Lax, 2002]. It would be worthwhile to explore that field to determine if our results can be rested on firmer foundations. In particular, exploration of the justification for the integration in (3.7.4) might permit an extension to allow us to place our GP on the logarithm of the likelihood q. This is desirable for at least a few reasons. Firstly, the log-likelihood is typically smoother than the likelihood and hence a better candidate for a GP. More importantly, this would allow us to enforce the non-negativity of q, better informing the GP and hence achieving better results.

In discriminating between a probability space and its logarithm, this argument has touched on the concept of the geometry of a probability space. One of the principal assumptions for the notable maximum entropy priors [Jaynes, 2003] is the notion that entropy provides a useful notion of separation in a probability space. This concept is generalised and considered more fully by *information geometry* [Amari and Nagaoka, 2000]. Information geometrical arguments are of interest not for manipulating probabilities, a niche completely occupied by probability theory. However, the field is immediately relevant wherever we need to explore or measure a probability space. It hence offers interesting insight into prior assignation, as exploited by entropic priors [Rodriguez, 1991, 2002, Caticha and Preuss, 2004]. Perhaps more relevantly, it also represents a more principled approach to the exploration required by any sampling method for quadrature of probabilities. The methods of information geometry may hence be exploited to perform truly optimal sampling, a welcome panacea to the myriad of heuristics currently employed.

However, easily the most important and extensive future work to be done is to apply our techniques to multi-agent systems. As mentioned earlier, the concept of second order probabilities is necessary in order to allow agents to have beliefs about what other agents might believe, and make decisions accordingly. Interestingly, it is proposed by Dennett [1991] that such modelling of other beings as agents lead historically to the development of consciousness in humans. Unfortunately, doing so in practice leads to a morass of nested 'I believe that he believes that I believe that . . .' integrals. This is essentially one of the problems faced by decentralised data fusion [Durrant-Whyte, 2002] and game theory [Fudenberg and Tirole, 1991]. Equally, even a single agent will be required to perform inference over all possible future times; it must effectively treat its future selves as distinct, unknown agents. For example, in order to determine the value of making an observation, an agent is required to marginalise all possible sets of information it could consequently possess in the future. Herein we encounter the problems of reinforcement learning [Sutton and Barto, 1998]. In order to resolve these undeniably difficult problems, we require powerful but principled approximation techniques. We believe Gaussian processes and BMC may help to fill this role.

A number of clear application suggest themselves. Primarily, we are interested in sensor networks. A clear generalisation of the work performed above is to treat each sensor as a full agent in its own right. Earlier, we gave each sensor a limited decision making capacity by allowing it to choose when next to take an observation and when to discard irrelevant data. However, we can imagine many other interesting decisions we might want agents to make - when and where to move in order to better observe a target, for example. Furthermore, all the work performed so far has effectively involved the compilation of the information arriving from all sensors by a single overseer-agent. It would be fascinating to consider a decentralised system, in which each agent must maintain its own individual beliefs, on the basis of what messages and observations it receives. Here we have further challenging decision problems - when and what should agents communicate?

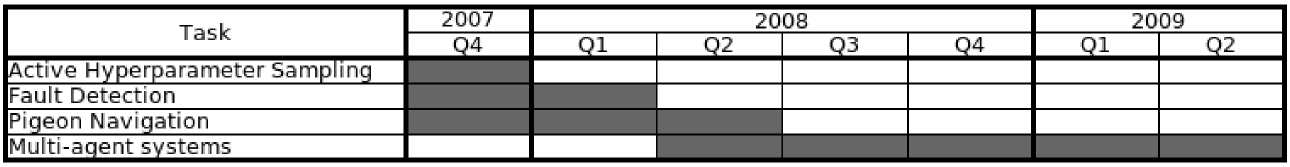
The remarkable capacity of pigeons to fly home, even from never before visited locations, remains an active area of biological research [Roberts et al., 2004]. As such, another interesting application for our methods is the tracking of pigeon paths, with a view to determining their methods of navigation. As a first step, we have already undertaken the modelling of pigeon paths as GPs, the resulting hyperparameters revealing something about the structure of how pigeons home. Unfortunately, space has precluded further description of our work in this report.

Even more tantalising, however, is the possibility of modelling pigeons as agents. Their utility appears well-defined - make it to the roost while using as little energy as possible. More interesting is the consideration of what information is available to these pigeon-agents, how they perceive their environment, what assumptions they implicitly make. No doubt evolution will have fitted pigeons with a set of assumptions that allow them to make best use of their limited computational ability. A full agent model of a pigeon might resolve many of the problems facing interested ornithologists.

In fact, there is nothing unique about software, humans or pigeons in our modelling of them as agents [Dennett, 1987]. The decisions made by these beings, the consequences of the motion of sub-atomic particles in their processors, are all equally determined by the underlying physical laws of the universe. All agents are necessarily physical bodies and are governed as such. Conversely, we are also fully entitled to model any body as an agent, taking the trajectory through the relevant decision space that maximises its relevant utility, given the information available to it. Here we have a suggestive analogy with variational (Lagrangian) mechanics [Landau and Lifshitz, 1960, Lanczos, 1986, Goldstein et al., 2002], in which bodies take the trajectory that extremises their action.

This approach offers particularly interesting insights for multiple agent problems. In particular, consider a team of agents co-operating to achieve a single goal, their decisions guided by a single utility. We draw an analogy with multiple particle systems, which are clearly able to resolve their trajectories through the extremising of an appropriate joint action. This resolution is mediated by what we call forces, the exchange of packets (virtual particles) of information. But if we consider this system as a collection of agents, these forces represent the messages passed in order to best achieve their joint goal - they represent the total information available to each particle-agent as it makes its decision of how to move. If we are to have a general theory of optimal message passing amongst agents, it must apply to this special case. Hence the forces we find acting between particles suggest the form of how any agents should best communicate, as determined by the laws of nature. It seems possible that further consideration may reveal efficient methods for message passing in the extensive literature of mechanics.

6.3 Schedule of Further Work



Task	2007	2008				2009	
	Q4	Q1	Q2	Q3	Q4	Q1	Q2
Active Hyperparameter Sampling	■						
Fault Detection	■	■					
Pigeon Navigation	■	■	■				
Multi-agent systems			■	■	■	■	■

Figure 6.1: Gantt chart of research schedule.

Appendix A

Identities

A.1 Gaussian Identities

The following identities [Rasmussen and Williams, 2006] are almost indispensible when dealing with Gaussian distributions, which we denote in the usual way as

$$\mathbf{N}(\boldsymbol{x};\, \boldsymbol{\mu},\, \mathbf{A}) \triangleq \frac{1}{\sqrt{\det 2\pi \mathbf{A}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\, \mathbf{A}^{-1}\,(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{A.1.1}$$

Probably the most important properties of a multivariate Gaussian distribution are that its marginals and conditionals are both themselves Gaussian. That is, if we have

$$p(\,\boldsymbol{x},\, \boldsymbol{y} \,|\, I\,) = \mathbf{N}\left(\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix};\begin{bmatrix}\boldsymbol{\mu}\\ \boldsymbol{\nu}\end{bmatrix},\begin{bmatrix}\mathbf{A} & \mathbf{C}\\ \mathbf{C}^{\mathrm{T}} & \mathbf{B}\end{bmatrix}\right) \tag{A.1.2}$$

then its marginal and conditional distributions are respectively:

$$p(\,\boldsymbol{x} \,|\, I\,) = \mathbf{N}(\boldsymbol{x};\, \boldsymbol{\mu},\, \mathbf{A}) \tag{A.1.3}$$

$$p(\,\boldsymbol{x} \,|\, \boldsymbol{y},\, I\,) = \mathbf{N}\big(\boldsymbol{x};\, \boldsymbol{\mu} + \mathbf{C}\,\mathbf{B}^{-1}(\boldsymbol{y}-\boldsymbol{\nu}),\, \mathbf{A} - \mathbf{C}\,\mathbf{B}^{-1}\,\mathbf{C}^{\mathrm{T}}\big) \tag{A.1.4}$$

Now, if we have

$$\begin{aligned} p(\,\boldsymbol{x} \,|\, I\,) &= \mathbf{N}(\boldsymbol{x};\, \boldsymbol{\mu},\, \boldsymbol{\Lambda}) \\ p(\,\boldsymbol{y} \,|\, \boldsymbol{x}\, I\,) &= \mathbf{N}(\boldsymbol{y};\, \mathbf{A}\,\boldsymbol{x} + \boldsymbol{b},\, \mathbf{L}) \end{aligned} \tag{A.1.5}$$

then the joint distribution can be written as

$$p(\,\boldsymbol{x},\, \boldsymbol{y} \,|\, I\,) = \mathbf{N}\left(\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix};\begin{bmatrix}\boldsymbol{\mu}\\ \mathbf{A}\,\boldsymbol{\mu}+\boldsymbol{b}\end{bmatrix},\begin{bmatrix}\boldsymbol{\Lambda} & \boldsymbol{\Lambda}\,\mathbf{A}^{\mathrm{T}}\\ \mathbf{A}\,\boldsymbol{\Lambda} & \mathbf{L}+\mathbf{A}\,\boldsymbol{\Lambda}\,\mathbf{A}^{\mathrm{T}}\end{bmatrix}\right) \tag{A.1.6}$$

and so, using (A.1.3) and (A.1.4)

$$p(\,\boldsymbol{y} \,|\, I\,) = \mathbf{N}\big(\boldsymbol{y};\, \mathbf{A}\,\boldsymbol{\mu} + \boldsymbol{b},\, \mathbf{L} + \mathbf{A}\,\boldsymbol{\Lambda}\,\mathbf{A}^{\mathrm{T}}\big) \tag{A.1.7}$$

$$p(\,\boldsymbol{x} \,|\, \boldsymbol{y},\, I\,) = \mathbf{N}(\boldsymbol{x};\, \boldsymbol{\mu} + \boldsymbol{\Gamma}(\boldsymbol{y} - \mathbf{A}\,\boldsymbol{\mu} - \boldsymbol{b}),\, \boldsymbol{\Lambda} - \boldsymbol{\Gamma}\,\mathbf{A}\,\boldsymbol{\Lambda}) \tag{A.1.8}$$

where

$$\mathbf{\Gamma} = \mathbf{\Lambda}\mathbf{A}^{\mathrm{T}}\left(\mathbf{L} + \mathbf{A}\,\mathbf{\Lambda}\,\mathbf{A}^{\mathrm{T}}\right)^{-1} \tag{A.1.9}$$

Similarly, note that a Gaussian distribution can be rewritten by applying the change of coordinates given by any unitary matrix $\mathbf{A}$

$$p(\,\boldsymbol{x} \mid I\,) = \mathbf{N}(\boldsymbol{x};\, \boldsymbol{\mu},\, \mathbf{\Lambda}) = \mathbf{N}\big(\mathbf{A}\,\boldsymbol{x};\, \mathbf{A}\,\boldsymbol{\mu},\, \mathbf{A}\,\mathbf{\Lambda}\,\mathbf{A}^{\mathrm{T}}\big) \tag{A.1.10}$$

A.2 Kronecker Identities

We use $\mathbf{I}_d$ to represent the identity matrix of dimension d. Similarly, $\mathbf{1}_{m,n}$ is a matrix containing all ones of dimensions $m \times n$; $\mathbf{1}_d$ is the square matrix of dimension d containing all ones. Now for a simple use of the mixed product property of the Kronecker product. Consider any matrix $\mathbf{A}$ with row (first) dimension Q (note that $\mathbf{A}$ could be a column vector of dimension Q), the mixed-product property of the Kronecker product gives:

$$\begin{aligned}\mathbf{1}_{P,1} \otimes \mathbf{A} &= (\mathbf{1}_{P,1}\,\mathbf{1}_{1,1}) \otimes (\mathbf{I}_Q\,\mathbf{A}) \\ &= (\mathbf{1}_{P,1} \otimes \mathbf{I}_Q)\,(\mathbf{1}_{1,1} \otimes \mathbf{A}) \\ &= (\mathbf{1}_{P,1} \otimes \mathbf{I}_Q)\,\mathbf{A} \end{aligned} \tag{A.2.1}$$

Similarly, it can be shown that for a matrix $\mathbf{B}$ with column (second) dimension Q

$$\mathbf{1}_{1,P} \otimes \mathbf{B} = \mathbf{B}\,(\mathbf{1}_{1,P} \otimes \mathbf{I}_Q) \tag{A.2.2}$$

Bibliography

P. Abrahamsen. A review of Gaussian random fields and correlation functions. Technical Report 917, Norwegian Computing Center, Box 114, Blindern, N-0314 Oslo, Norway, 1997. URL http://www.math.ntnu.no/~omre/TMA4250/V2007/abrahamsen2.ps. 2nd edition.

S. Amari and H. Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical Monographs*. Oxford University Press, 2000.

P. Boyle and M. Frean. Dependent Gaussian processes. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 217–224. MIT Press, Cambridge, MA, 2005.

G. L. Bretthorst. The near-irrelevance of sampling frequency distributions. In W. von der Linden et al., editor, *Maximum Entropy and Bayesian Methods*, pages 21–46. Kluwer Academic Publishers, the Netherlands, 1999.

A. Caticha and R. Preuss. Maximum entropy and Bayesian data analysis: entropic priors. *Physical Review E*, 70:046127, 2004. URL http://arXiv.org/physics/0307055.

T. Y.-K. Chan. CHIMET: Weather reports from Chichester Bar, 2000. URL http://www.chimet.co.uk.

R. T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13, 1946. URL http://link.aip.org/link/?AJP/14/1/1.

B. de Finetti. La prévision: Ses lois logiques, ses sources subjectives. In *Annales de l'Institut Henri Poincaré 7*, pages 1–68. Paris, 1937. Translated into English by Henry E. Kyburg Jr., Foresight: Its Logical Laws, its Subjective Sources. In Henry E. Kyburg Jr. and Howard E. Smokler (1964, Eds.), Studies in Subjective Probability, 53-118, Wiley, New York.

B. de Finetti. Probabilities of probabilities: a real problem or a misunderstanding? In B. C. Aykac A, editor, *New developments in the application of Bayesian methods*, pages 1–10, 1977.

D. C. Dennett. *Consciousness Explained*. Little, Brown and Company, Boston, New York, 1991.

D. C. Dennett. *The Intentional Stance*. Cambridge: MIT Press, 1987.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, second edition, November 2000. ISBN 0471056693.

H. Durrant-Whyte. Multi sensor data fusion. Technical report, Australian Centre for Field Robotics, The University of Sydney, NSW, January 2001.

H. Durrant-Whyte. Introduction to decentralised data fusion. Technical report, Australian Centre for Field Robotics, The University of Sydney, NSW, September 2002.

D. Fudenberg and J. Tirole. *Game theory*. MIT Press, Cambridge, MA, 1991.

M. N. Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, Cambridge University, 1997. URL http://www.inference.phy.cam.ac.uk/mng10/GP/thesis.ps.

A. Girard, C. Rasmussen, J. Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs – application to multiple-step ahead time series forecasting. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2003.

T. Gneiting. Compactly supported correlation functions. *Journal of Multivariate Analysis*, 83:493–508, 2002.

R. Goldsmith and N.-E. Sahlin. The role of second-order probabilities in decision making. In P. Humphreys, O. Svenson, and A. Vari, editors, *Analysing and Aiding Decision Processes*, North-Holland, Amsterdam, 1983.

H. Goldstein, C. Poole, and J. Safko. *Classical Mechanics*. Addison Wesley, third edition, 2002.

P. Gregory. *Bayesian Logical Data Analysis for the Physical Sciences: A Comparative Approach with Mathematica Support*. Cambridge University Press, April 2005.

D. Heath and W. Sudderth. De Finetti's Theorem on exchangeable variables. *The American Statistician*, 30(4):188–189, November 1976.

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

H. Jeffreys. *Theory of Probability*. Oxford University Press, third edition, November 1998.

K. H. Knuth. Lattice duality: The origin of probability and entropy. *Neurocomputing*, 67:245–274, 2005.

C. Lanczos. *The variational principles of mechanics*. Dover publications, Inc., New York, fourth edition, 1986.

L. Landau and E. M. Lifshitz. *Mechanics, Course of Theoretical Physics, Volume 1*. Pergamon Press, 1960.

P. D. Lax. *Functional Analysis*. Wiley-Interscience, New York, 2002.

T. J. Loredo. Computational technology for Bayesian inference. In D. M. Mehringer, R. L. Plante, and D. A. Roberts, editors, *ASP Conference Series 172: Astronomical Data Analysis Software and Systems*, volume 8, pages 297–, San Francisco, 1999.

D. J. MacKay. The humble Gaussian distribution, June 2006. URL http://www.inference.phy.cam.ac.uk/mackay/humble.ps.gz.

D. J. MacKay. Introduction to Gaussian processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, pages 84–92. Springer-Verlag, 1998. URL ftp://www.inference.phy.cam.ac.uk/pub/mackay/gpB.ps.gz.

D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, June 2002. ISBN 0521642981.

J. Meade, D. Biro, and T. Guilford. Homing pigeons develop local route stereotypy. *Proceedings of the Royal Society B: Biological Sciences*, 272:17–23, January 2005.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993. URL http://www.cs.toronto.edu/~radford/ftp/review.pdf.

R. M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report 9702, Dept. of Statistics, University of Toronto, 1997. URL http://www.cs.toronto.edu/~radford/ftp/mc-gp.pdf.

A. O'Hagan. Bayes-hermite quadrature. *Journal of Statistical Planning and Inference*, 29:245–260, 1991.

A. O'Hagan. Some Bayesian numerical analysis. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 4*, pages 345–363. Oxford University Press, 1992.

A. O'Hagan. Monte Carlo is fundamentally unsound. *The Statistician*, 36:247–249, 1987.

J. Pinheiro and D. Bates. Unconstrained parameterizations for variance-covariance matrices. *Statistics and Computing*, 6:289–296, 1996.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992. ISBN 0521437148.

C. E. Rasmussen. Gaussian processes to speed up hybrid monte carlo for expensive Bayesian integrals. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, editors, *Bayesian Statistics 7*, pages 651–659. Oxford University Press, 2003.

C. E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. In S. T. S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, Cambridge, MA, 2003.

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

S. Roberts, T. Guilford, I. Rezek, and D. Biro. Positional entropy during pigeon homing. I. Application of Bayesian latent state modelling. *J. Theor. Biol*, 227(1):39–50, 2004.

C. C. Rodriguez. Entropic priors for discrete probabilistic networks and for mixtures of Gaussians models. In R. L. Fry, editor, *Bayesian Inference and Maximum Entropy Methods in Science and Engineering, AIP Conf. Proc. 617*, APL Johns Hopkins University, August 2002. URL http://arxiv.org/abs/physics/0201016.

C. C. Rodriguez. Entropic priors, 1991. URL http://omega.albany.edu:8008/entpriors.ps.

J. Skilling. Nested sampling for Bayesian computations. In *Proceedings of the Valencia / ISBA 8th World Meeting on Bayesian Statistics*, Benidorm (Alicante, Spain), June 2006.

M. L. Stein. *Interpolation of Spatial Data*. Springer Verlag, New York, 1999.

R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1998. ISBN 0262193981.

The MathWorks. MATLAB R2007a, 2007. Natick, MA.

J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In *Advances in Neural Information Processing Systems 18*, pages 1441–1448. The MIT Press, 2006. Proc. NIPS'05.