Bayesian Inference: An Introduction to Principles and Practice in Machine Learning

Michael E. Tipping
Microsoft Research, Cambridge, U.K.

Published as:	"Bayesian inference: An introduction to Principles and practice in machine learning." In O. Bousquet, U. von Luxburg, and G. Rätsch (Eds.), Advanced Lectures on Machine Learning, pp. 41–62. Springer.
Year of publication:	2004
This version typeset:	June 26, 2006
Available from:	`http://www.miketipping.com/papers.htm`
Correspondence:	`mail@miketipping.com`

Abstract This article gives a basic introduction to the principles of Bayesian inference in a machine learning context, with an emphasis on the importance of marginalisation for dealing with uncertainty. We begin by illustrating concepts via a simple regression task before relating ideas to practical, contemporary, techniques with a description of 'sparse Bayesian' models and the 'relevance vector machine'.

1 Introduction

What is meant by "Bayesian inference" in the context of machine learning? To assist in answering that question, let's start by proposing a conceptual task: we wish to learn, from some given number of example instances of them, a model of the relationship between pairs of variables A and B. Indeed, many machine learning problems are of the type "given A, what is B?".[1]

Verbalising what we typically treat as a mathematical task raises an interesting question in itself. How do we answer "what is B?"? Within the appealingly well-defined and axiomatic framework of propositional logic, we 'answer' the question with complete certainty, but this logic is clearly too rigid to cope with the realities of real-world modelling, where uncertaintly over 'truth' is ubiquitous. Our measurements of both the dependent (B) and independent (A) variables are inherently noisy and inexact, and the relationships between the two are invariably non-deterministic. This is where probability theory comes to our aid, as it furnishes us with a principled and consistent framework for meaningful reasoning in the presence of uncertainty.

We might think of probability theory, and in particular Bayes' rule, as providing us with a "logic of uncertainty" [1]. In our example, given A we would 'reason' about the likelihood of the truth of B (let's say B is binary for example) via its conditional probability $P(B|A)$: that is, "what is the probability of B given that A takes a particular value?". An appropriate answer might be "B is true with probability 0.6". One of the primary tasks of 'machine learning' is then to approximate $P(B|A)$ with some appropriately specified model based on a given set of corresponding examples of A and B.[2]

[1] In this article we will focus exclusively on such 'supervised learning' tasks, although of course there are other modelling applications which are equally amenable to Bayesian inferential techniques.

[2] In many learning methods, this conditional probability approximation is not made explicit, though such an interpretation may exist. However, one might consider it a significant limitation if a particular machine learning procedure *cannot* be expressed coherently within a probabilistic framework.

It is in the modelling procedure where Bayesian inference comes to the fore. We typically (though not exclusively) deploy some form of parameterised model for our conditional probability:

$$P(B|A) = f(A; \mathbf{w}), \tag{1}$$

where $\mathbf{w}$ denotes a vector of all the 'adjustable' parameters in the model. Then, given a set $\mathcal{D}$ of N examples of our variables, $\mathcal{D} = \{A_n, B_n\}_{n=1}^{N}$, a conventional approach would involve the maximisation of some measure of 'accuracy' (or minimisation of some measure of 'loss') of our model for $\mathcal{D}$ with respect to the adjustable parameters. We then can make predictions, given A, for unknown B by evaluating $f(A; \mathbf{w})$ with parameters $\mathbf{w}$ set to their optimal values. Of course, if our model f is made too complex — perhaps there are many adjustable parameters $\mathbf{w}$ — we risk over-specialising to the observed data $\mathcal{D}$, and consequently realising a poor model of the true underlying distribution $P(B|A)$.

The first key element of the Bayesian inference paradigm is to treat parameters such as $\mathbf{w}$ as random variables, exactly the same as A and B. So the conditional probability now becomes $P(B|A, \mathbf{w})$, and the dependency of the probability of B on the parameter settings, as well as A, is made explicit. Rather than 'learning' comprising the optimisation of some quality measure, a *distribution* over the parameters $\mathbf{w}$ is inferred from Bayes' rule. We will demonstrate this concept by means of a simple example regression task in Section 2.

To obtain this 'posterior' distribution over $\mathbf{w}$ alluded to above, it is necessary to specify a 'prior' distribution $p(\mathbf{w})$ before we observe the data. This may be considered an incovenience, but Bayesian inference treats all sources of uncertainty in the modelling process in a unified and consistent manner, and forces us to be explicit as regards our assumptions and constraints; this in itself is arguably a philosophically appealing feature of the paradigm.

However, the most attractive facet of a Bayesian approach is the manner in which "Ockham's Razor" is automatically implemented by 'integrating out' all irrelevant variables. That is, under the Bayesian framework there is an automatic preference for simple models that sufficiently explain the data without unnecessary complexity. We demonstrate this key feature in Section 3, and in particular underline the point that *this property holds even if the prior* $p(\mathbf{w})$ *is completely uninformative*. We show that, in practical terms, the concept of Ockham's Razor enables us to 'set' regularisation parameters and 'select' models without the need for any additional validation procedure.

The practical disadvantage of the Bayesian approach is that it requires us to perform integrations over variables, and many of these computations are analytically intractable. As a result, much contemporary research in Bayesian approaches to machine learning relies on, or is directly concerned with, approximation techniques. However, we show in Section 4, where we describe the "sparse Bayesian" model, that a combination of analytic calculation and straightforward, practically efficient, approximation can offer state-of-the-art results.

2 From Least-Squares to Bayesian Inference

We introduce the methodology of Bayesian inference by considering an example prediction (regression) problem. Let us assume we are given a very simple data set (illustrated later within Figure 1) comprising $N = 15$ samples artificially generated from the function $y = \sin(x)$ with added Gaussian noise of variance 0.2. We will denote the 'input' variables in our example by x_n, $n = 1 \ldots N$. For each such x_n, there is an associated real-valued 'target' t_n, $n = 1 \ldots N$, and from these input-target pairs, we wish to 'learn' the underlying functional mapping.

2.1 Linear Models

We will model this data with some parameterised function $y(x;\mathbf{w})$, where $\mathbf{w} = (w_1, w_2, \ldots, w_M)$ is the vector of adjustable model parameters. Here, we consider linear models (strictly, "linear-in-the-parameter") models which are a linearly-weighted sum of M fixed (but potentially *non*linear) basis functions $\phi_m(x)$:

$$y(x;\mathbf{w}) = \sum_{m=1}^{M} w_m \phi_m(x). \tag{2}$$

For our purposes here, we make the common choice to utilise Gaussian data-centred basis functions $\phi_m(x) = \exp\left\{-(x-x_m)^2/r^2\right\}$, which gives us a 'radial basis function' (RBF) type model.

2.1.1 "Least-squares" Approximation.

Our objective is to find values for $\mathbf{w}$ such that $y(x;\mathbf{w})$ makes good predictions for new data: *i.e.* it models the *underlying generative function*. A classic approach to estimating $y(x;\mathbf{w})$ is "least-squares", minimising the error measure:

$$E_{\mathcal{D}}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left[t_n - \sum_{m=1}^{M} w_m \phi_m(x_n)\right]^2. \tag{3}$$

If $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$ and $\mathbf{\Phi}$ is the 'design matrix' such that $\mathbf{\Phi}_{nm} = \phi_m(x_n)$, then the minimiser of (3) is obtained in closed-form via linear algebra:

$$\mathbf{w}_{LS} = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}. \tag{4}$$

However, with $M = 15$ basis functions and only $N = 15$ examples here, we know that minimisation of squared-error leads to a model which exactly interpolates the data samples, as shown in Figure 1.

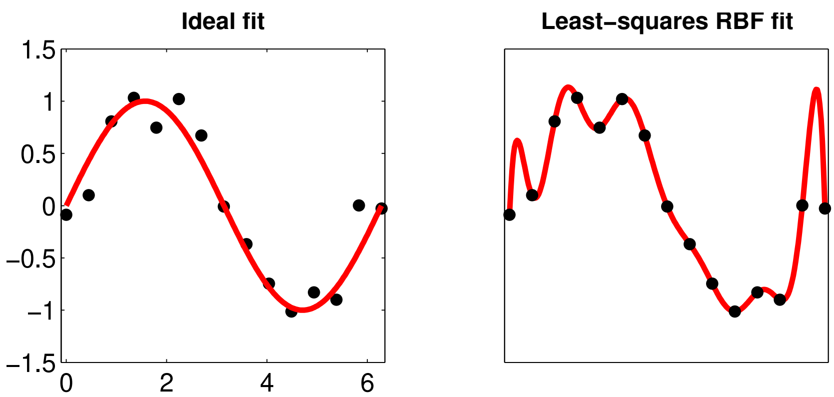

Figure 1: Overfitting? The 'ideal fit' is shown on the left, while the least-squares fit using 15 basis functions is shown on the right and perfectly interpolates all the data points.

Now, we may look at Figure 1 and exclaim "the function on the right is clearly over-fitting!". But, without prior knowledge of the 'truth', can we really judge which model is genuinely better? The answer is that we can't — in a real-world problem, the data could quite possibly have been generated by a complex function such as shown on the right. The only way that we can proceed to meaningfully learn from data such as this is by imposing some *a priori* prejudice on the nature of the complexity of functions we expect to elucidate. A common way of doing this is via 'regularisation'.

2.2 Complexity Control: Regularisation

A common, and generally very reasonable, assumption is that we typically expect that data is generated from smooth, rather than complex, functions. In a linear model framework, smoother functions typically have smaller weight magnitudes, so we can penalise complex functions by adding an appropriate penalty term to the cost function that we minimise:

$$\widehat{E}(\mathbf{w}) = E_{\mathcal{D}}(\mathbf{w}) + \lambda E_W(\mathbf{w}). \tag{5}$$

A standard choice is the squared-weight penalty, $E_W(\mathbf{w}) = \frac{1}{2}\sum_{m=1}^{M} w_m^2$, which conveniently gives the "penalised least-squares" (PLS) estimate for $\mathbf{w}$:

$$\mathbf{w}_{PLS} = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \lambda\mathbf{I})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}. \tag{6}$$

The *hyperparameter* λ balances the trade-off between $E_{\mathcal{D}}(\mathbf{w})$ and $E_W(\mathbf{w})$ — *i.e.* between how well the function fits the data and how smooth it is. Given that we can compute the weights directly for a given λ, the learning problem is now transformed into one of finding an appropriate value for that hyperparameter. A very common approach is to assess potential values of λ according to the error calculated on a set of 'validation' data (*i.e.* data which is not used to estimate $\mathbf{w}$), and examples of fits for different values of λ and their associated validation errors are given in Figure 2.

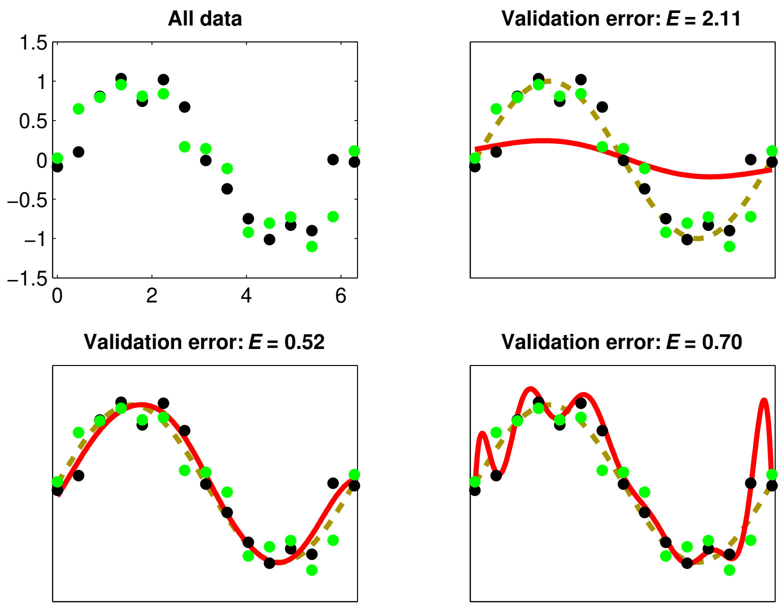

Figure 2: Function estimates (solid line) and validation error for three different values of regularisation hyperparameter λ (the true function is shown dashed). The training data is plotted in black, and the validation set in green (gray).

In practice, we might evaluate a large number of models with different hyperparameter values and select the model with lowest validation error, as demonstrated in Figure 3. We would then hope that this would give us a model which was close to 'the truth'. In this artificial case where we know the generative function, the deviation from 'truth' is illustrated in the figure with the measurement of 'test error', the error on noise-free samples of $\sin(x)$. We can see that the minimum validation error does not quite localise the best test error, but it is arguably satisfactorily close. We'll come back to this graph in Section 3 when we look at marginalisation and how Bayesian inference can be exploited in order to estimate λ. For now, we look at how this regularisation approach can be initially reformulated within a Bayesian probabilistic framework.

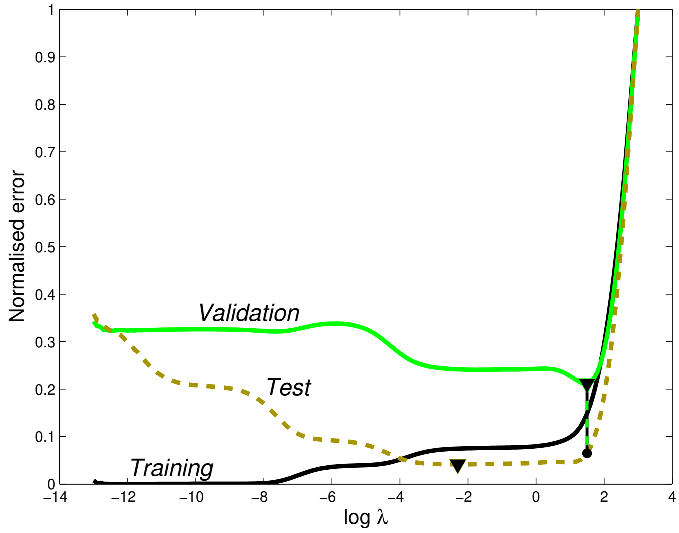

Figure 3: Plots of error computed on the separate 15-example training and validation sets, along with 'test' error measured on a third noise-free set. The minimum test and validation errors are marked with a triangle, and the intersection of the best λ computed via validation is shown.

2.3 A Probabilistic Regression Framework

We assume as before that the data is a noisy realisation of an underlying functional model: $t_n = y(x_n; \mathbf{w}) + \epsilon_n$. Applying least-squares resulted in us minimising $\sum_n \epsilon_n^2$, but here we first define an explicit probabilistic model over the noise component ϵ_n, chosen to be a Gaussian distribution with mean zero and variance σ^2. That is, $p(\epsilon_n|\sigma^2) = N(0, \sigma^2)$. Since $t_n = y(x_n; \mathbf{w}) + \epsilon_n$ it follows that $p(t_n|x_n,\mathbf{w}, \sigma^2) = N(y(x_n; \mathbf{w}), \sigma^2)$. Assuming that each example from the the data set has been generated independently (an often realistic assumption, although not always true), the *likelihood*[3] of all the data is given by the product:

$$p(\mathbf{t}|\mathbf{x},\mathbf{w}, \sigma^2) = \prod_{n=1}^{N} p(t_n|x_n,\mathbf{w}, \sigma^2), \tag{7}$$

$$= \prod_{n=1}^{N} (2\pi\sigma^2)^{-1/2} \exp\left[-\frac{\{t_n - y(x_n; \mathbf{w})\}^2}{2\sigma^2}\right]. \tag{8}$$

Note that, from now on, we will write terms such as $p(\mathbf{t}|\mathbf{x},\mathbf{w}, \sigma^2)$ as $p(\mathbf{t}|\mathbf{w}, \sigma^2)$, since we never seek to model the given input data $\mathbf{x}$. Omitting to include such conditioning variables is purely for notational convenience (it implies no further model assumptions) and is common practice.

2.4 Maximum Likelihood and Least-Squares

The 'maximum-likelihood' estimate for $\mathbf{w}$ is that value which maximises $p(\mathbf{t}|\mathbf{w}, \sigma^2)$. In fact, this is identical to the 'least-squares' solution, which we can see by noting that minimising squared-error

[3] Although 'probabilty' and 'likelihood' functions may be identical, a common convention is to refer to "probability" when it is primarily interpreted as a function of the random variable $\mathbf{t}$, and "likelihood" when interpreted as a function of the parameters $\mathbf{w}$.

is equivalent to minimising the negative logarithm of the likelihood which here is:

$$-\log p(\mathbf{t}|\mathbf{w},\sigma^2) = \frac{N}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{n=1}^{N}\{t_n - y(x_n;\mathbf{w})\}^2 . \tag{9}$$

Since the first term on the right in (9) is independent of $\mathbf{w}$, this leaves only the second term which is proportional to the squared error.

2.5 Specifying a Bayesian Prior

Of course, giving an identical solution for $\mathbf{w}$ as least-squares, maximum likelihood estimation will also result in overfitting. To control the model complexity, instead of the earlier regularisation weight penalty $E_W(\mathbf{w})$, we now define a *prior distribution* which expresses our 'degree of belief' over values that $\mathbf{w}$ might take:

$$p(\mathbf{w}|\alpha) = \prod_{m=1}^{M}\left(\frac{\alpha}{2\pi}\right)^{1/2}\exp\left\{-\frac{\alpha}{2}w_m^2\right\}. \tag{10}$$

This (common) choice of a zero-mean Gaussian prior, expresses a preference for smoother models by declaring smaller weights to be *a priori* more probable. Though the prior is independent for each weight, there is a shared inverse variance hyperparameter α, analogous to λ earlier, which moderates the strength of our 'belief'.

2.6 Posterior Inference

Previously, given our error measure and regulariser, we computed a single *point estimate* $\mathbf{w}_{LS}$ for the weights. Now, given the likelihood and the prior, we compute the *posterior distribution* over $\mathbf{w}$ via Bayes' rule:

$$p(\mathbf{w}|\mathbf{t},\alpha,\sigma^2) = \frac{\text{likelihood} \times \text{prior}}{\text{normalising factor}} = \frac{p(\mathbf{t}|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha)}{p(\mathbf{t}|\alpha,\sigma^2)}. \tag{11}$$

As a consequence of combining a Gaussian prior and a linear model within a Gaussian likelihood, the posterior is also conveniently Gaussian: $p(\mathbf{w}|\mathbf{t},\alpha,\sigma^2) = N(\boldsymbol{\mu},\boldsymbol{\Sigma})$ with

$$\boldsymbol{\mu} = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \sigma^2\alpha\mathbf{I})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}, \tag{12}$$

$$\boldsymbol{\Sigma} = \sigma^2(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \sigma^2\alpha\mathbf{I})^{-1}. \tag{13}$$

So instead of 'learning' a single value for $\mathbf{w}$, we have inferred a distribution over all possible values. In effect, we have updated our prior 'belief' in the parameter values in light of the information provided by the data $\mathbf{t}$, with more posterior probability assigned to values which are both probable under the prior and which 'explain the data'.

2.6.1 MAP Estimation: a 'Bayesian' Short-cut.

The "maximum *a posteriori*" (MAP) estimate for $\mathbf{w}$ is the single most probable value under the posterior distribution $p(\mathbf{w}|\mathbf{t},\alpha,\sigma^2)$. Since the denominator in Bayes' rule (11) earlier is independent of $\mathbf{w}$, this is equivalent to maximising the numerator, or equivalently minimising $E_{MAP}(\mathbf{w}) = -\log p(\mathbf{t}|\mathbf{w},\sigma^2) - \log p(\mathbf{w}|\alpha)$. Retaining only those terms dependent on $\mathbf{w}$ gives:

$$E_{MAP}(\mathbf{w}) = \frac{1}{2\sigma^2}\sum_{n=1}^{N}\{t_n - y(x_n;\mathbf{w})\}^2 + \frac{\alpha}{2}\sum_{m=1}^{M}w_m^2. \tag{14}$$

The MAP estimate is therefore identical to the PLS estimate with $\lambda = \sigma^2\alpha$.

2.6.2 Illustration of Sequential Bayesian Inference.

For our example problem, we'll end this section by looking at how the posterior $p(\mathbf{w}|\mathbf{t}, \alpha, \sigma^2)$ evolves as we observe increasingly more data points t_n. Before proceeding, we note that we can compute the posterior incrementally since here the data are assumed independent (conditioned on $\mathbf{w}$). *e.g.* for $\mathbf{t} = (t_1, t_2, t_3)$:

$$\begin{aligned} p(\mathbf{w}|t_1, t_2, t_3) &\propto p(t_1, t_2, t_3|\mathbf{w})\ p(\mathbf{w}), \\ &= p(t_2, t_3|\mathbf{w})\ p(t_1|\mathbf{w})\ p(\mathbf{w}), \\ &= \text{Likelihood of } (t_2, t_3) \times \text{posterior having observed } t_1. \end{aligned}$$

So, more generally, we can treat the posterior having observed $(t_1, \ldots, t_K)$ as the 'prior' for the remaining data $(t_{K+1}, \ldots, t_N)$ and obtain the equivalent result to seeing all the data at once. We exploit this result in Figure 4 where we illustrate how the posterior distribution updates with increasing amounts of data.

The second row in Figure 4 illustrates some relevant points. First, because the data observed up to that point are not generally near the centres of the two basis functions visualised, those values of x are relatively uninformative regarding the associated weights and the posterior thereover has not deviated far from the prior. Second, on the far right in the second row, we can see that the function is fairly well determined in the vicinity of the observations, but at higher values of x, where data are yet to be observed, the MAP estimate of the function is not accurate and the posterior samples there exhibit high variance. On the third row we have observed all data, and notice that although the MAP predictor appears subjectively good, the posterior still seems quite diffuse and the variance in the samples is noticable. We emphasise this point in the bottom row, where we have generated and observed an extra 200 data points and it can be seen how the posterior is now much more concentrated, and samples from it are now quite closely concentrated about the MAP value.

Note that this facility to sample from the prior or posterior is a very informative feature of the Bayesian paradigm. For the posterior, it is a helpful way of visualising the remaining uncertainty in parameter estimates in cases where the posterior distribution itself cannot be visualised. Furthermore, the ability to visualise samples from the prior alone is very advantageous, as it offers us evidence to judge the appropriateness of our prior assumptions. No equivalent facility exists within the regularisation or penalty function framework.

3 Marginalisation and Ockham's Razor

Since we have just seen that the *maximum a posteriori* (MAP) and penalised least-squares (PLS) estimates are equivalent, it might be tempting to assume that the Bayesian framework is simply a probabilistic re-interpretation of classical methods. This is certainly not the case! It is sometimes overlooked that the distinguishing element of Bayesian methods is really *marginalisation*, where instead of seeking to 'estimate' all 'nuisance' variables in our models, we attempt to integrate them out. As we will now see, this is a powerful component of the Bayesian framework.

3.1 Making Predictions

First lets reiterate some of the previous section and consider how, having 'learned' from the training values $\mathbf{t}$, we make a prediction for the value of t_* given a new input datum x_*:

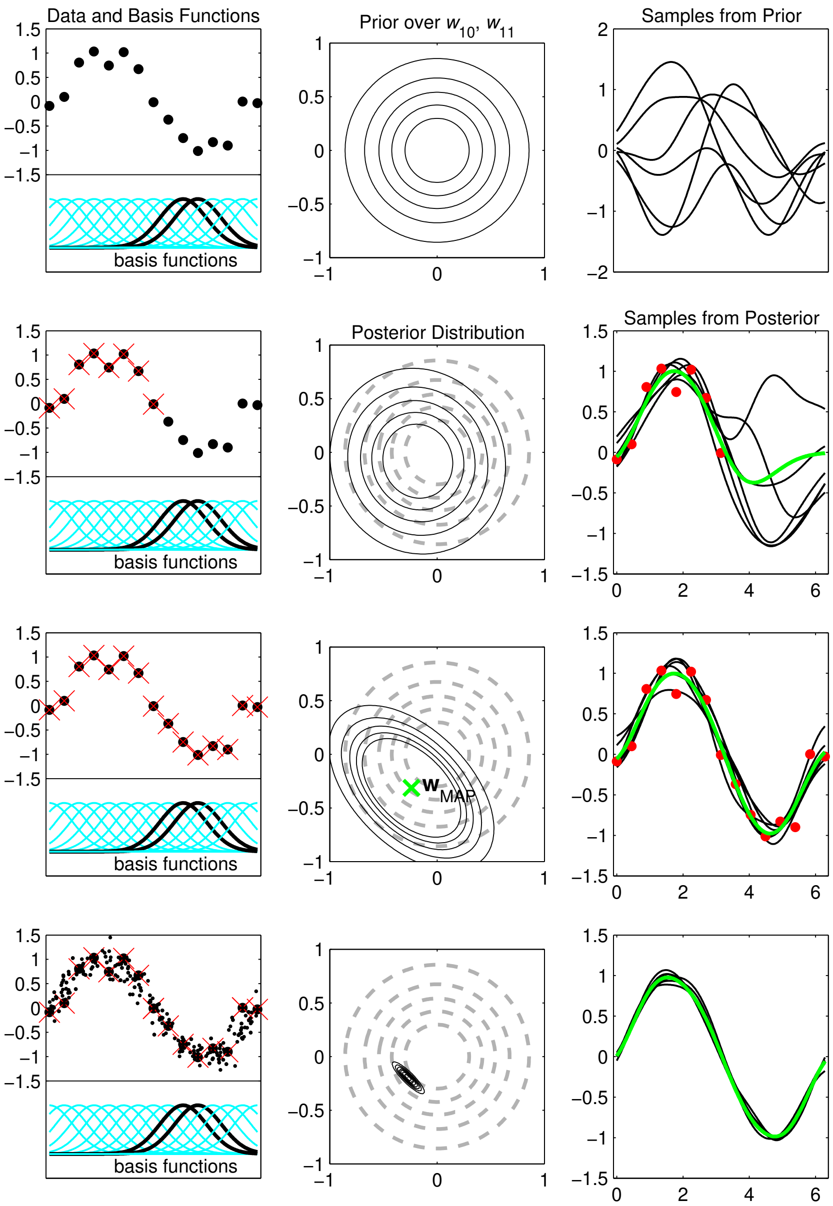

Figure 4: Illustration of the evolution of the posterior distribution as data is sequentially 'absorbed'. The left column shows the data, with those points which have been observed so far crossed, along with a plot of the basis functions. The contour plots in the middle column show the prior/posterior over just two (for visualisation purposes) of the weights, w_{10} and w_{11}, corresponding to the highlighted basis functions on the left. The right hand column plots $y(x; \mathbf{w})$ from a number of samples of $\mathbf{w}$ from the full prior/posterior, along with the posterior mean, or MAP, estimator (in thicker green/gray). From top to bottom, the number of data is increasing. Row 1 shows the *a priori* case for no data, row 2 shows the model after 8 examples, and row 3 shows the model after all 15 data points have been observed. Finally, the bottom row shows the case when an additional 200 data points have been generated and absorbed in the posterior model.

Framework	Learned Quantity	Prediction
Classical	$\mathbf{w}_{PLS}$	$y(x_*; \mathbf{w}_{PLS})$
MAP Bayesian	$p(\mathbf{w}\|\mathbf{t}, \alpha, \sigma^2)$	$p(t_*\|\mathbf{w}_{MAP}, \sigma^2)$
True Bayesian	$p(\mathbf{w}\|\mathbf{t}, \alpha, \sigma^2)$	$p(t_*\|\mathbf{t}, \alpha, \sigma^2)$

The first two approaches result in similar predictions, although the MAP Bayesian model does give a probability distribution for t_* (which can be sampled from, *e.g.* see Figure 4). The mean of this distribution is the same as that of the classical predictor $y(x_*; \mathbf{w}_{PLS})$, since $\mathbf{w}_{MAP} = \mathbf{w}_{PLS}$.

However, the 'true Bayesian' way is to *integrate out*, or *marginalise* over, the uncertain variables $\mathbf{w}$ in order to obtain the *predictive distribution*:

$$p(t_*|\mathbf{t}, \alpha, \sigma^2) = \int p(t_*|\mathbf{w}, \sigma^2)\ p(\mathbf{w}|\mathbf{t}, \alpha, \sigma^2)\ d\mathbf{w}. \tag{15}$$

This distribution $p(t_*|\mathbf{t}, \alpha, \sigma^2)$ incorporates our uncertainty over the weights having seen $\mathbf{t}$, by averaging the model probability for t_* over all possible values of $\mathbf{w}$. If we are unsure about the parameter settings, for example if there were very few data points, then $p(\mathbf{w}|\mathbf{t}, \alpha, \sigma^2)$ and similarly $p(t_*|\mathbf{t}, \alpha, \sigma^2)$ will be appropriately diffuse. The classical, and even MAP Bayesian, predictions take no account of how well-determined our parameters $\mathbf{w}$ really are.

3.2 The General Bayesian Predictive Framework

You way well find the presence of α and σ^2 as conditioning variables in the predictive distribution, $p(t_*|\mathbf{t}, \alpha, \sigma^2)$, in (15) rather disconcerting, and indeed, for any general model, if we wish to predict t_* given some training data $\mathbf{t}$, what we really, really want is $p(t_*|\mathbf{t})$. That is, we wish to integrate out *all* variables not directly related to the task at hand. So far, we've only placed a prior over the weights $\mathbf{w}$ — to be truly, truly Bayesian, we should define $p(\alpha)$, a so-called *hyperprior*, along with a prior over the noise level $p(\sigma^2)$. Then the full posterior over 'nuisance' variables becomes:

$$p(\mathbf{w}, \alpha, \sigma^2|\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)p(\alpha)p(\sigma^2)}{p(\mathbf{t})}. \tag{16}$$

The denominator, or normalising factor, in (16) is the *marginalised* probability of the data:

$$p(\mathbf{t}) = \int p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)p(\alpha)p(\sigma^2)\ d\mathbf{w}\ d\alpha\ d\sigma^2, \tag{17}$$

and is nearly always analytically intractable to compute! Nevertheless, as we'll soon see, $p(\mathbf{t})$ is a very useful quantity and can be amenable to effective approximation.

3.3 Practical Bayesian Prediction

Given the full posterior (16), Bayesian inference in our example regression model would proceed with:

$$p(t_*|\mathbf{t}) = \int p(t_*|\mathbf{w}, \sigma^2)\ p(\mathbf{w}, \alpha, \sigma^2|\mathbf{t})\ d\mathbf{w}\ d\alpha\ d\sigma^2, \tag{18}$$

but as we indicated, we can't compute either $p(\mathbf{w}, \alpha, \sigma^2|\mathbf{t})$ or $p(t_*|\mathbf{t})$ analytically. If we wish to proceed, we must turn to some approximation strategy (and it is here that much of the Bayesian "voodoo" resides). A sensible approach might be to perform those integrations that are analytically computable, and then approximate remaining integrations, perhaps using one of a number of established methods:

- Type-II maximum likelihood (discussed shortly)
- Laplace's method (see, *e.g.*, [2])
- Variational techniques (see, *e.g.*, [3, 4])

- Sampling (*e.g.* [2, 5])

Much research in Bayesian inference has gone, and continues to go, into the development and assessment of approximation techniques, including those listed above. For the purposes of this article, we will primarily exploit the first of them.

3.4 A Type-II Maximum Likelihood Approximation

Here, using the product rule of probability, we can rewrite the ideal full posterior $p(\mathbf{w}, \alpha, \sigma^2|\mathbf{t})$ as:

$$p(\mathbf{w}, \alpha, \sigma^2|\mathbf{t}) \equiv p(\mathbf{w}|\mathbf{t}, \alpha, \sigma^2)\ p(\alpha, \sigma^2|\mathbf{t}). \tag{19}$$

The first term is our earlier weight posterior which we have already computed: $p(\mathbf{w}|\mathbf{t}, \alpha, \sigma^2) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The second term $p(\alpha, \sigma^2|\mathbf{t})$ we will approximate, admittedly crudely, by a δ-function at its mode. *i.e.* we find "most probable" values α_{MP} and σ^2_{MP} which maximise:

$$p(\alpha, \sigma^2|\mathbf{t}) = \frac{p(\mathbf{t}|\alpha, \sigma^2)\ p(\alpha)\ p(\sigma^2)}{p(\mathbf{t})}. \tag{20}$$

Since the denominator is independent of α and σ^2, we only need maximise the numerator $p(\mathbf{t}|\alpha, \sigma^2)p(\alpha)p(\sigma^2)$. Furthermore, if we assume flat, *uninformative*, priors over $\log\alpha$ and $\log\sigma$, then we equivalently just need to find the maximum of $p(\mathbf{t}|\alpha, \sigma^2)$. Assuming a flat prior here may seem to be a computational convenience, but in fact it is arguably our prior of choice since our model will be invariant to the scale of the target data (and basis set), which is almost always an advantageous feature[4]. For example, our results won't change if we measure t in metres instead of miles. We'll return to the task of maximising $p(\mathbf{t}|\alpha, \sigma^2)$ in Section 3.6.

3.5 The Approximate Predictive Distribution

Having found α_{MP} and σ^2_{MP}, our approximation to the predictive distribution would be:

$$\begin{aligned} p(t_*|\mathbf{t}) &= \int p(t_*|\mathbf{w}, \sigma^2)\ p(\mathbf{w}|\mathbf{t}, \alpha, \sigma^2)\ p(\alpha, \sigma^2|\mathbf{t})\ d\mathbf{w}\ d\alpha\ d\sigma^2, \\ &\approx \int p(t_*|\mathbf{w}, \sigma^2)\ p(\mathbf{w}|\mathbf{t}, \alpha, \sigma^2)\ \delta(\alpha_{\mathrm{MP}}, \sigma^2_{\mathrm{MP}})\ d\mathbf{w}\ d\alpha\ d\sigma^2, \\ &= \int p(t_*|\mathbf{w}, \sigma^2_{\mathrm{MP}})\ p(\mathbf{w}|\mathbf{t}, \alpha_{\mathrm{MP}}, \sigma^2_{\mathrm{MP}})\ d\mathbf{w}. \end{aligned} \tag{21}$$

In our example earlier, recall that $p(\mathbf{w}|\mathbf{t}, \alpha_{\mathrm{MP}}, \sigma^2_{\mathrm{MP}}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, from which the approximate predictive distribution can be finally written as:

$$p(t_*|\mathbf{t}) \approx \int p(t_*|\mathbf{w}, \sigma^2_{\mathrm{MP}})\ p(\mathbf{w}|\mathbf{t}, \alpha_{\mathrm{MP}}, \sigma^2_{\mathrm{MP}})\ d\mathbf{w}. \tag{22}$$

This is now computable and is Gaussian: $N(\mu_*, \sigma^2_*)$, with:

$$\begin{aligned} \mu_* &= y(x_*; \boldsymbol{\mu}), \\ \sigma^2_* &= \sigma^2_{\mathrm{MP}} + \mathbf{f}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{f}, \end{aligned}$$

where $\mathbf{f} = [\phi_1(x_*), \ldots, \phi_M(x_*)]^{\mathrm{T}}$. Intuitively, we see that

[4] Note that for *scale* parameters such as α and σ^2, it can be shown that it is appropriate to define uniformative priors uniformly over a *logarithmic* scale [6]. While for brevity we will continue to denote parameters "α" and "σ", from now on we will work with the logarithms thereof, and in particular, will maximise distributions with respect to $\log\alpha$ and $\log\sigma$. In this respect, one must note with caution that finding the maximum of a distribution with respect to parameters is *not* invariant to transformations of those parameters, whereas the result of integration with respect to transformed distributions *is* invariant.

- the mean predictor μ_* is the model function evaluated with the posterior mean weights (the same as the MAP prediction),
- the predictive variance σ_*^2 is the sum of variances associated with both the noise process and the uncertainty of the weight estimates. In particular, it can be clearly seen that when the posterior over $\mathbf{w}$ is more diffuse, and $\mathbf{\Sigma}$ is larger, σ_*^2 is also increased.

3.6 Marginal Likelihood

Returning now to the question of finding α_{MP} and σ^2_{MP}, as noted ealier we find the maximising values of the 'marginal likelihood' $p(\mathbf{t}|\alpha, \sigma^2)$. This is given by:

$$\begin{aligned} p(\mathbf{t}|\alpha,\sigma^2) &= \int p(\mathbf{t}|\mathbf{w},\sigma^2)\; p(\mathbf{w}|\alpha)\; d\mathbf{w}, \\ &= (2\pi)^{-N/2}|\sigma^2\mathbf{I} + \alpha^{-1}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}|^{-1/2} \exp\left\{-\frac{1}{2}\mathbf{t}^{\mathrm{T}}(\sigma^2\mathbf{I}+\alpha^{-1}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}})^{-1}\mathbf{t}\right\}. \end{aligned} \tag{23}$$

This is a Gaussian distribution over the single N-dimensional dataset vector $\mathbf{t}$, and (23) is readily evaluated for arbitrary values of α (and σ^2). Note that here we can use *all the data* to directly determine α_{MP} and σ^2_{MP} — we don't need to reserve a separate data set to validate their values. We can use gradient-based techniques to maximise (23) (and we will do so for a similar quantity in Section 4), but here we choose to repeat the earlier experiment for the regularised linear model. While we fix σ^2 (though we could also experimentally evaluate it), in Figure 5 we have computed the marginal likelihood (in fact, its negative logarithm) at a number of different values of α (for just the 15-example training set, though we could also have made use of the validation set too) and compared with the training, validation and test errors of Figure 3 earlier.

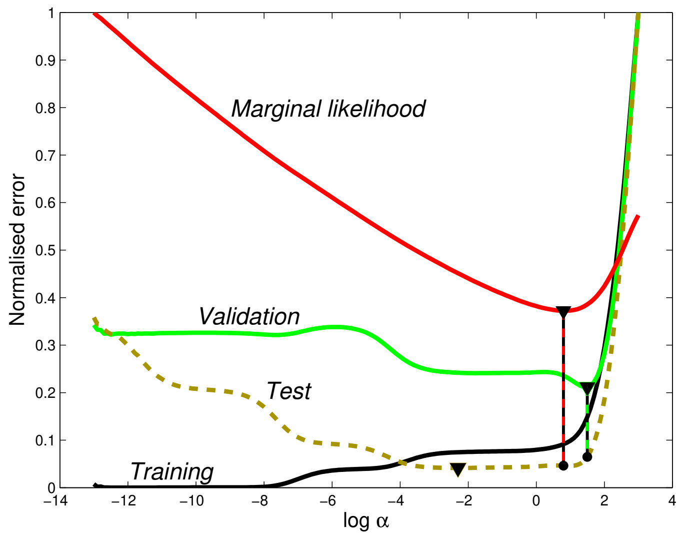

Figure 5: Plots of the training, validation and test errors of the model as shown in Figure 3 (with the horizontal scale adjusted appropriately to convert from λ to α) along with the negative log marginal likelihood evaluated *on the training data alone* for that same model. The values of α and test error achieved by the model with highest marginal likelihood (smallest negative log) are indicated.

It is quite striking that *using only 15 examples and no validation data*, the Bayesian approach for setting α (giving test error 1.66) finds a closer model to the 'truth' than the classical model with its validated value of λ (test error 2.33). It is also interesting to see, and is it not immediately obvious why, that the marginal likelihood measure, although only measured on the training data, is not monotonic (unlike training error) and exhibits a maximum at some intermediate complexity level. The marginal likelihood criterion appears to be successfully penalising *both* models that are too simple *and* too complex — this is "Ockham's Razor" at work.

3.7 Ockham's Razor

In the fourteenth century, William of Ockham proposed:

> "*Pluralitas non est ponenda sine neccesitate*"

which literally translates as "entities should not be multiplied unnecessarily". Its original historic context was theological, but the concept remains relevant for machine learning today, where it might be translated as "models should be no more complex than is sufficient to explain the data". The Bayesian procedure is effectively implementing "Ockham's Razor" by assigning lower probability *both* to models that are too simple *and* too complex. We might ask: why is an intermediate value of α preferred? The schematic of Figure 6 shows how this can be the case, as a result of the marginal likelihood $p(\mathbf{t}|\alpha)$ being a normalised distribution over the space of all possible data sets $\mathbf{t}$. Models with high α only fit (assign significant marginal probability to) data from smooth functions. Models with low values of α can fit data generated from functions that are both smooth and complex. However, because of normalisation, the low-α model must generally assign lower probability to data from smooth functions, so the marginal likelihood naturally prefers the simpler model if the data is smooth, which is precisely the meaning of Ockham's Razor. Furthermore, one can see from Figure 6 that for a data set of 'intermediate' complexity, a 'medium' value of α can be preferred. This is qualitatively analogous to the case of our example set, where we indeed find that an intermediate value of α is optimal. Note, crucially, that this is achieved without any prior preference for any particular value of α as we originally assumed a uniform hyperprior over its logarithm. The effect of Ockham's Razor is an automatic and pleasing consequence of applying the Bayesian framework.

3.8 Model Selection

While we have concentrated so far on the search for an appropriate value of hyperparameter α (and, to an extent, σ^2), our model is also conditioned on other variables we have up to now overlooked: the choice of basis set $\mathbf{\Phi}$ and, for our Gaussian basis, its width parameter r (as defined in Section 2.1). Ideally, we should define priors $P(\mathbf{\Phi})$ and $p(r)$, and integrate out those variables when making predictions. More practically, we could use $p(\mathbf{t}|\mathbf{\Phi}, r)$ as a criterion for *model selection* with the expectation that Ockham's Razor will assist us in selecting a model that is sufficient to explain the data but is not over-complex. In our example model, we previously optimised the marginal likelihood to find a value for α. In fact, as there are only two nuisance parameters here, it is feasible to integrate out α and σ^2 numerically.

In Figure 7 we evaluate several basis sets $\mathbf{\Phi}$ and width values r by computing the integral

$$p(\mathbf{t}|\mathbf{\Phi}, r) = \int p(\mathbf{t}|\alpha, \sigma^2, \mathbf{\Phi}, r)\ p(\alpha)\ p(\sigma^2)\ d\alpha\ d\sigma^2, \tag{24}$$

$$\approx \frac{1}{S}\sum_{s=1}^{S} p(\mathbf{t}|\alpha_s, \sigma_s^2, \mathbf{\Phi}, r), \tag{25}$$

with a Monte-Carlo average where we obtain S samples log-uniformly from $\alpha \in [10^{-12}, 10^{12}]$ and $\sigma \in [10^{-4}, 10^0]$.

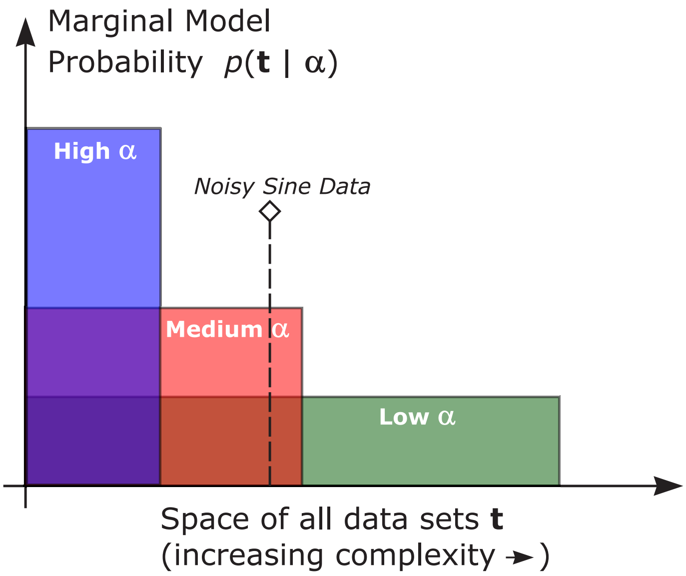

Figure 6: A schematic plot of three marginal probability distributions for 'high', 'medium' and 'low' values of α. The figure is a simplification of the case for the actual distribution $p(\mathbf{t}|\alpha)$, where for illustrative purposes the N-dimensional space of $\mathbf{t}$ has been compressed onto a single axis and where, notionally, data sets (instances of $\mathbf{t}$) arising from simpler (smoother) functions lie towards the left-hand end of the horizontal scale, and data from complex functions to the right.

The results of Figure 7 are quite compelling: with uniform priors over all nuisance variables —*i.e.* we have imposed *absolutely no prior knowledge* — we observe that test error appears very closely related to marginal likelihood. The qualitative shapes of the curves, and the relative merits, of Gaussian and Laplacian basis functions are also captured. For the Gaussian basis we are very close to obtaining the optimal value of r, in terms of test error, from just 15 examples and no validation data. Reassuringly, the simplest model that contains the 'truth', $y = w_1 \sin(x)$, is the most probable model here. We also show in the figure the model $y = w_1 \sin(x) + w_2 \cos(x)$ which is also an ideal fit for the data, but it is penalised in marginal probability terms since the addition of the $w_2 \cos(x)$ term allows it to explain more data sets, and normalisation thus requires it to assign less probability to our particular set. Nevertheless, it is still some orders of magnitude more probable than the Gaussian basis model.

3.9 Summary So Far...

Marginalisation is the key element of Bayesian inference, and hopefully some of the examples above have persuaded the reader that it can be an exceedingly powerful one. Problematically though, ideal Bayesian inference proceeds by integrating out all irrelevant variables, and we must concede that

- for practical purposes, it may be appropriate to require point estimates of some 'nuisance' variables, since it could easily be impractical to average over many parameters and particularly models every time we wish to make a prediction (imagine, for example, running a handwriting recogniser on a portable computing device),
- many of the desired integrations necessitate some form of approximation.

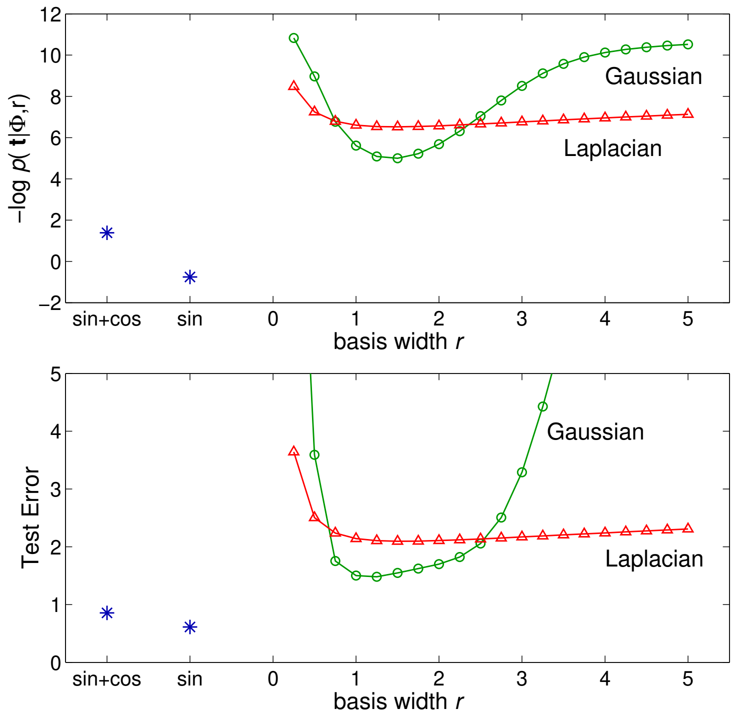

Figure 7: Top: negative log model probability $-\log p(\mathbf{t}|\mathbf{\Phi}, r)$ for various basis sets, evaluated by analytic integration over $\mathbf{w}$ and Monte-Carlo averaging over α and σ^2. **Bottom:** corresponding test error for the posterior mean predictor. Basis sets examined were 'Gaussian', $\exp\{-|x - x_m|^2/r^2\}$, 'Laplacian', $\exp\{-|x - x_m|/r\}$, $\sin(x)$, $\sin(x)$ with $\cos(x)$. For the Gaussian and Laplacian basis, the horizontal axis denotes varying 'width' parameter r shown. For the sine/cosine bases, the horizontal axis has no significance and the values are placed to the left for convenience.

Nevertheless, regarding these points, we can still leverage Bayesian techniques to considerable benefit exploiting carefully-applied approximations. In particular, marginalised likelihoods within the Bayesian framework allow us to estimate fixed values of hyperparameters where desired and, most beneficially, choose between models and their varying parameterisations. This can all be done without the need to use validation data. Furthermore:

- it is straightforward to estimate other parameters in the model that may be of interest, *e.g.* the noise variance,
- we can sample from both prior and posterior models of the data,
- the exact parameterisation of the model is irrelevant when integrating out,
- we can incorporate other priors of interest in a principled manner.

We now further demonstrate these points, notably the last one, in the next section where we present a practical framework for the inference of 'sparse' models.

4 Sparse Bayesian Models

4.1 Bayes and Contemporary Machine Learning

In the previous section we saw that marginalisation is a valuable component of the Bayesian paradigm which offers a number of advantageous features applicable to many data modelling tasks. Disadvantageously, we also saw that the integrations required for full Bayesian inference can often be analytically intractable, although approximations for simple linear models could be very effective. Historically, interest in Bayesian "machine learning" (but not statistics!) has focussed on approximations for *non-linear* models, *e.g.* for neural networks, the "evidence procedure" [7] and "hybrid Monte Carlo" sampling [5]. More recently, flexible (*i.e.* many-parameter) linear kernel methods have attracted much renewed interest, thanks mainly to the popularity of the "support vector machine". These kind of models, of course, are particularly amenable to Bayesian techniques.

4.1.1 Linear Models and Sparsity.

Much interest in linear models has focused on *sparse* learning algorithms, which set many weights w_m to zero in the estimated predictor function $y(x) = \sum_m w_m \phi_m(x)$. Sparsity is an attractive concept; it offers elegant complexity control, feature extraction, the potential for elucidation of meaningful input variables along with the practical benefits of computational speed and compactness.

How do we impose a preference for sparsity in a model? The most common approach is via an appropriate regularisation term or prior. The most common regularisation term that we have already met, $E_W(\mathbf{w}) = \sum_{m=1}^{M} |w_m|^2$, of course corresponds to a Gaussian prior and is easy to work with, but while it is an effective way to control complexity, it does not promote sparsity. In the regularisation sense, the 'correct' term would be $E_W(\mathbf{w}) = \sum_m |w_m|^0$, but this, being discontinuous in w_m, is very difficult to work with. Instead, $E_W(\mathbf{w}) = \sum_m |w_m|^1$ is a workable compromise which gives reasonable sparsity and reasonable tractability, and is exploited in a number of methods, including as a Laplacian prior $p(\mathbf{w}) \propto \exp(-\sum_m |w_m|)$ [8]. However, there is an arguably more elegant way of obtaining sparsity within a Bayesian framework that builds effectively on the ideas outlined in the previous section and we conclude this article with a brief outline thereof.

4.2 A Sparse Bayesian Prior

In fact, we *can* obtain sparsity by retaining the traditional Gaussian prior, which is great news for tractability. The modification to our earlier Gaussian prior (10) is subtle:

$$p(\mathbf{w}|\alpha_1, \ldots, \alpha_M) = \prod_{m=1}^{M} \left[(2\pi)^{-1/2} {\alpha_m}^{1/2} \exp\left\{ -\frac{1}{2} \alpha_m w_m^2 \right\} \right]. \tag{26}$$

In constrast to the model in Section 2, we now have M hyperparameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$, one α_m independently controlling the (inverse) variance of each weight w_m.

4.2.1 A Hierarchical Prior.

The prior $p(\mathbf{w}|\boldsymbol{\alpha})$ is nevertheless still Gaussian, and superficially seems to have little preference for sparsity. However, it remains conditioned on $\boldsymbol{\alpha}$, so for full Bayesian consistency we should now define hyperpriors over all α_m. Previously, we utilised a log-uniform hyperprior — this is a special case of a *Gamma* hyperprior, which we introduce for greater generality here. This combination of the prior over α_m controlling the prior over w_m gives us what is often referred to as a *hierarchical* prior. Now, if we have $p(w_m|\alpha_m)$ and $p(\alpha_m)$ and we want to know the 'true' $p(w_m)$ we already know what to do — we must marginalise:

$$p(w_m) = \int p(w_m|\alpha_m)\ p(\alpha_m)\ d\alpha_m. \tag{27}$$

For a Gamma $p(\alpha_m)$, this integral is computable and we find that $p(w_m)$ is a *Student-t* distribution illustrated as a function of two parameters in Figure 8; its equivalent as a regularising penalty function would be $\sum_m \log |w_m|$.

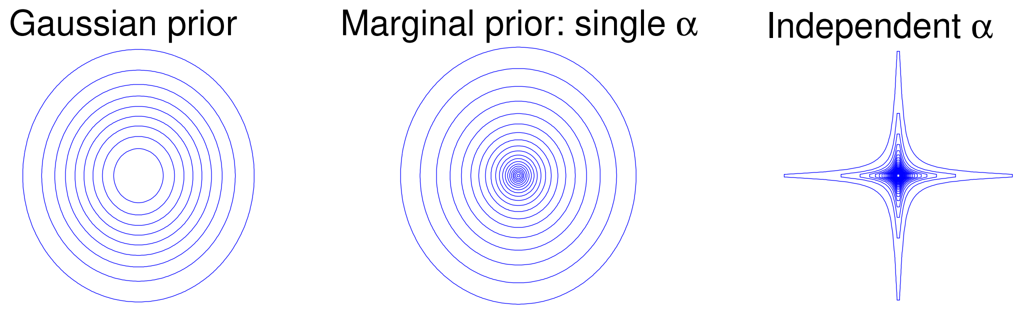

Figure 8: Contour plots of Gaussian and Student-t prior distributions over two parameters. While the marginal prior $p(w_1, w_2)$ for the 'single' hyperparameter model of Section 2 has a much sharper peak than the Gaussian at zero, it can be seen that it is not sparse unlike the multiple 'independent' hyperparameter prior, which as well as having a sharp peak at zero, places most of its probability mass along axial ridges where the magnitude of one of the two parameters is small.

4.3 A Sparse Bayesian Model for Regression

We can develop a sparse regression model by following an identical methodology to the previous sections. Again, we assume independent Gaussian noise: $t_n \sim N(y(\mathbf{x}_n; \mathbf{w}), \sigma^2)$, which gives a corresponding likelihood:

$$p(\mathbf{t}|\mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{t} - \mathbf{\Phi}\mathbf{w}\|^2\right\}, \tag{28}$$

where as before we denote $\mathbf{t} = (t_1 \ldots t_N)^{\mathrm{T}}$, $\mathbf{w} = (w_1 \ldots w_M)^{\mathrm{T}}$, and $\mathbf{\Phi}$ is the $N \times M$ 'design' matrix with $\mathbf{\Phi}_{nm} = \phi_m(\mathbf{x}_n)$.

Following the Bayesian framework, we desire the posterior distribution over all unknowns:

$$p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2|\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{w}, \boldsymbol{\alpha}, \sigma^2)p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2)}{p(\mathbf{t})}, \tag{29}$$

which we can't compute analytically. So as previously, we decompose this as:

$$p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2|\mathbf{t}) \equiv p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2)\ p(\boldsymbol{\alpha}, \sigma^2|\mathbf{t}) \tag{30}$$

where $p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2)$ is the 'weight posterior' distribution, and is tractable. This leaves $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{t})$ which must be approximated.

4.3.1 The Weight Posterior Term.

Given the data, the posterior distribution over weights is Gaussian:

$$
\begin{aligned}
p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\sigma^2) &= \frac{p(\mathbf{t}|\mathbf{w},\sigma^2)\,p(\mathbf{w}|\boldsymbol{\alpha})}{p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)}, \\
&= (2\pi)^{-(N+1)/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\left\{-\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{w}-\boldsymbol{\mu})\right\}, \qquad (31)
\end{aligned}
$$

with

$$
\boldsymbol{\Sigma} = (\sigma^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}+\mathbf{A})^{-1}, \qquad (32)
$$

$$
\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}, \qquad (33)
$$

and where we collect all the hyperparameters into a diagonal matrix: $\mathbf{A} = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_M)$. A key point to note from (31–33) is that if any $\alpha_m = \infty$, the corresponding $\mu_m = 0$.

4.3.2 The Hyperparameter Posterior Term.

Again we will adopt the "type-II maximum likelihood" approximation where we maximise $p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)$ to find $\boldsymbol{\alpha}_{\mathrm{MP}}$ and σ^2_{MP}. As before, for uniform hyperpriors over $\log\alpha$ and $\log\sigma$, $p(\boldsymbol{\alpha},\sigma^2|\mathbf{t}) \propto p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)$, where the *marginal likelihood* $p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)$ is obtained by integrating out the weights:

$$
\begin{aligned}
p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2) &= \int p(\mathbf{t}|\mathbf{w},\sigma^2)\,p(\mathbf{w}|\boldsymbol{\alpha})\;d\mathbf{w}, \\
&= (2\pi)^{-N/2}|\sigma^2\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}|^{-1/2}\exp\left\{-\frac{1}{2}\mathbf{t}^{\mathrm{T}}(\sigma^2\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{\mathrm{T}})^{-1}\mathbf{t}\right\}. \qquad (34)
\end{aligned}
$$

In Section 2, we found the single α_{MP} empirically but here for multiple (in practice, perhaps thousands of) hyperparameters, we cannot experimentally explore the space of possible $\boldsymbol{\alpha}$ so we instead optimise $p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)$ directly, via a gradient-based approach.

4.3.3 Hyperparameter Re-estimation.

Differentiating $\log p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)$ with respect to α and σ^2, setting to zero and rearranging (see [9]) ultimately gives iterative re-estimation formulae:

$$
\alpha_i^{\mathrm{new}} = \frac{\gamma_i}{\mu_i^2}, \qquad (35)
$$

$$
(\sigma^2)^{\mathrm{new}} = \frac{\|\mathbf{t}-\boldsymbol{\Phi}\boldsymbol{\mu}\|^2}{N-\sum_{i=1}^{M}\gamma_i}. \qquad (36)
$$

For convenience we have defined

$$
\gamma_i = 1 - \alpha_i\boldsymbol{\Sigma}_{ii}, \qquad (37)
$$

where $\gamma_i \in [0,1]$ is a measure of 'well-determinedness' of parameter w_i. This quantity effectively captures the influence of the likelihood (total when $\gamma \to 1$) and the prior (total when $\gamma \to 0$) on the value of each w_i. Note that the quantities on the right-hand-side of equations (35–37) are computed using the 'old' values of $\boldsymbol{\alpha}$ and σ^2.

4.3.4 Summary of Inference Procedure.

We're now in a position to define a 'learning algorithm' for approximate Bayesian inference in this model:

1. Initialise all $\{\alpha_i\}$ and σ^2 (or fix latter if known)
2. Compute weight posterior sufficient statistics $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$
3. Compute all $\{\gamma_i\}$, then re-estimate $\{\alpha_i\}$ (and σ^2 if desired)
4. Repeat from 2. until convergence
5. 'Delete' weights (and basis functions) for which optimal $\alpha_i = \infty$, since this implies $\mu_i = 0$
6. Make predictions for new data via the predictive distribution computed with the converged $\boldsymbol{\alpha}_{\mathrm{MP}}$ and σ^2_{MP}:
$$p(t_*|\mathbf{t}) = \int p(t_*|\mathbf{w}, \sigma^2_{\mathrm{MP}})\; p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}_{\mathrm{MP}}, \sigma^2_{\mathrm{MP}})\; d\mathbf{w} \tag{38}$$
the mean of which is $y(\mathbf{x}_*; \boldsymbol{\mu})$

Step 5. rather ideally assumes that we can reliably estimate such large values of α, whereas in reality limited computational precision implies that in this algorithm we have to place some finite upper limit on α (*e.g.* 10^{12} times the value of the smallest α). In many real-world tasks, we do indeed find that many α_i do tend towards infinity, and we converge toward a model that is very sparse, even if M is very large.

4.4 The "Relevance Vector Machine" (RVM)

To give an example of the potential of the above model, we briefly introduce here the "Relevance Vector Machine" (RVM), which is simply a specialisation of a sparse Bayesian model which utilises the same data-dependent kernel basis as the popular "support vector machine" (SVM):

$$y(\mathbf{x}; \mathbf{w}) = \sum_{n=1}^{N} w_n K(\mathbf{x}, \mathbf{x}_n) + w_0 \tag{39}$$

This model is described, with a number of examples, in much more detail elsewhere [9]. For now, Figure 9 provides an illustration, on some noise-polluted synthetic data, of the potential of this Bayesian framework for effectively combining sparsity with predictive accuracy.

5 Summary

While the tone of the first three sections of this article has been introductory and the models considered therein have been quite simplistic, the brief example of the 'sparse Bayesian' learning procedure given in Section 4 is intended to demonstrate that 'practical' Bayesian inference procedures have the potential to be highly effective in the context of modern machine learning. Readers who find this demonstration sufficiently convincing and who are interested specifically in the sparse Bayesian model framework can find further information (including some implementation code), and details of related approaches, at a web-page maintained by the author: `http://www.research.microsoft.com/mlp/RVM/`. In particular, note that the algorithm for hyperparameter estimation of Section 4.3.3 was presented here as it has a certain intuitive simplicity, but in fact there is a much more efficient and practical approach to optimising $\log p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)$ which is detailed in [10].

We summarised some of the features, advantages and limitations of the general Bayesian framework earlier in Section 3.9, and so will not repeat them here. The reader interested in investigating further and in more depth on this general topic may find much helpful further material in the references [1, 5, 11, 12, 13, 14].

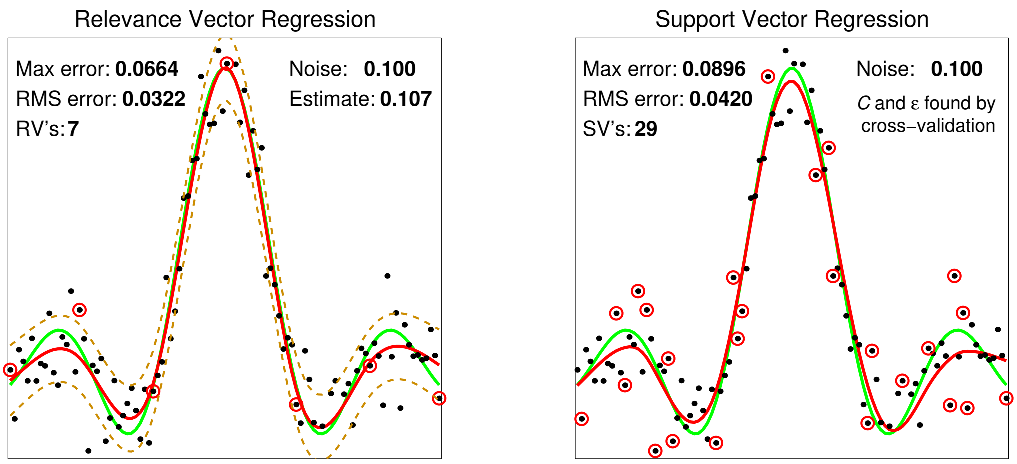

Figure 9: The relevance vector and support vector machines applied to a regression problem using a Gaussian kernel, which demonstrates some of the advantages of the Bayesian approach. Of particular note is the sparsity of the final Bayesian model, which qualitatively appears near-optimal. It is also worth underlining that the 'nuisance' parameters C and ϵ for the SVM had to be found by a separate cross-validation procedure, whereas the RVM algorithm estimates them automatically, and arguably quite accurately in the case of the noise variance.

References

[1] Edward T. Jaynes. *Probability theory: the logic of science*. Cambridge University Press, 2003.

[2] M Evans and T B Swartz. Methods for approximating integrals in statistics with special emphasis on Bayesian integration. *Statistical Science*, 10(3):254–272, 1995.

[3] Matt Beal and Zoubin Ghahramani. The Variational Bayes web site at `http://www.variational-bayes.org/`, 2003.

[4] Christopher M Bishop and Michael E Tipping. Variational relevance vector machines. In Craig Boutilier and Moisés Goldszmidt, editors, *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 46–53. Morgan Kaufmann, 2000.

[5] Radford M Neal. *Bayesian Learning for Neural Networks*. Springer, 1996.

[6] James O Berger. *Statistical decision theory and Bayesian analysis*. Springer, second edition, 1985.

[7] David J C MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992.

[8] Peter M Williams. Bayesian regularisation and pruning using a Laplace prior. *Neural Computation*, 7(1):117–143, 1995.

[9] Michael E Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[10] Michael E. Tipping and Anita C Faul. Fast marginal likelihood maximisation for sparse Bayesian models. In C. M. Bishop and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Key West, FL, Jan 3-6*, 2003.

[11] David J C MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[12] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[13] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian Data Analysis*. Chapman & Hall, 1995.

[14] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.