3.10 CHAPTER SUMMARY

This chapter has provided an introduction to the Bayesian way of performing Machine Learning tasks – treating all parameters as random variables. We have performed a Baysesian analysis for a coin tossing model and the linear regression model introduced in Chapters 1 and 2. In both cases, we defined prior densities over parameters, defined likelihoods and computed posterior densities. In both examples, the prior and likelihood were chosen such that the posterior could be computed analytically. In addition, we computed predictions by taking expectations with respect to the posterior and introduced marginal likelihood as a possible model selection criterion.

Unfortunately, these expressions are not often analytically tractable and we must resort to sampling and approximation techniques. These techniques are the foundations of modern Bayesian inference and form an important area of Machine Learning research and development. The next chapter will describe three popular techniques – point estimates, Laplace approximations and Markov chain Monte Carlo.

3.11 EXERCISES

3.1 For $\alpha, \beta = 1$, the beta distribution becomes uniform between 0 and 1. In particular, if the probability of a coin landing heads is given by r and a beta prior is placed over r, with parameters $\alpha = 1$, $\beta = 1$, this prior can be written as

$$p(r) = 1 \quad (0 \leq r \leq 1).$$

Using this prior, compute the posterior density for r if y heads are observed in N tosses (i.e. multiply this prior by the binomial likelihood and manipulate the result to obtain something that looks like a beta density).

3.2 Repeat the previous exercise for the following prior, also a particular form of the beta density:

$$p(r) = \begin{cases} 2r & 0 \leq r \leq 1 \\ 0 & \textbf{otherwise} \end{cases}$$

What are the values of the prior parameters α and β that result in $p(r) = 2r$?

3.3 Repeat the previous exercise for the following prior (again, a form of beta density):

$$p(r) = \begin{cases} 3r & 0 \leq r \leq 1 \\ 0 & \textbf{otherwise.} \end{cases}$$

What are the prior parameters here?

3.4 What are the effective prior sample sizes (α and β) for the previous three exercises (i.e. how many heads and tails are they equivalent to)?

3.5 If a random variable R has a beta density

$$p(r) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1}(1-r)^{\beta-1},$$

derive an expression for the expected value of r, $\mathbf{E}_{p(r)}\{r\}$. You will need the following identity for the gamma function:

$$\Gamma(n+1) = n\Gamma(n)$$

Hint: Use the fact that

$$\int_{r=0}^{r=1} r^{\alpha-1}(1-r)^{b-1}dr = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}.$$

3.6 Using the setup in the previous exercise, and the identity

$$\mathrm{var}\Big\{r\Big\} = \mathbf{E}_{p(r)}\Big\{r^2\Big\} - \left(\mathbf{E}_{p(r)}\{r\}\right)^2,$$

derive an expression for var{*r*}. You will need the gamma identity given in the previous exercise.

3.7 At a different stall, you observe 20 tosses of which 9 were heads. Compute the posteriors for the three scenarios, the probability of winning in each case and the marginal likelihoods.

3.8 Use MATLAB to generate coin tosses where the probability of heads is 0.7. Generate 100 tosses and compute the posteriors for the three scenarios, the probabilities of winning and the marginal likelihoods.

3.9 In Section 3.8.4 we derived an expression for the Gaussian posterior for a linear model within the context of the Olympic 100m data. Substituting $\mu_0 = [0, 0, \ldots, 0]^{\mathsf{T}}$, we saw the similarity between the posterior mean

$$\boldsymbol{\mu}_{\mathbf{w}} = \tfrac{1}{\sigma^2}\left(\tfrac{1}{\sigma^2}\mathbf{X}^{\mathrm{T}}\mathbf{X} + \sum_0^{-1}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}.$$

and the regularised least squares solution

$$\widehat{\mathbf{w}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + N \diagdown \mathbf{I}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}.$$

For this particular example, find the prior covariance matrix $\boldsymbol{\Sigma}_0$ that makes the two identical. In other words, find $\boldsymbol{\Sigma}_0$ in terms of λ.

3.10 Redraw the graphical representation of the Olympic 100m model to reflect the fact that the prior over $\mathbf{w}$ is actually conditioned on $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$.

3.11 In Figure 3.25 we studied the effect of reducing $\boldsymbol{\sigma_0^2}$ on the marginal likelihood. Using MATLAB, investigate the effect of increasing $\boldsymbol{\sigma_0^2}$.

3.12 When performing a Bayesian analysis on the Olympics data, we assumed that the prior was known. If a Gaussian prior is placed on $\mathbf{w}$ and an inverse gamma prior on the variance σ^2

$$p(\sigma^2 \Big| \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}(\sigma^2)^{-\alpha-1}\exp\left\{-\frac{\beta}{\sigma^2}\right\},$$

the posterior will also be the product of a Gaussian and an inverse gamma. Compute the posterior parameters.

3.12 FURTHER READING

[1] Ben Calderhead and Mark Girolami. Estimating Bayes factors via thermodynamic integration and population MCMC. *Comput. Stat. Data Anal.*, 53:4028–4045, October 2009.

An article by the authors describing a novel approach for calculating the marginal likelihoods (Bayes factors) in models where it is not analytically tractable.

[2] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, second edition, 2004.

One of the most popular textbooks on Bayesian inference. Provides a detailed and practical description of Bayesian Inference.

[3] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision*, pages 343–356, 1996.

An interesting example of the use of Bayesian methods in the field of human computer interaction. The authors use a sampling technique to infer posterior probabilities over gestures being performed by users.

[4] Michael Jordan, editor. *Learning in Graphical Models*. MIT Press, 1999.

An introduction to the field of graphical models and how to use them for learning tasks.

[5] Christian Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer, second edition edition, 2007.

[6] Tian-Rui Xu et al. Inferring signaling pathway topologies from multiple perturbation measurement of specific biochemical species. *Science Signalling*, 3(113), 2010.

A paper showing how Bayesian model selection via the marginal likelihood can be used to answer interesting scientific questions in the field of biology. It is also an interesting example of large-scale Bayesian sampling.

CHAPTER 4

Bayesian Inference

In the previous chapter we introduced the key concepts required to adopt a Bayesian approach to machine learning. Within the Bayesian framework, all unknown quantities are treated as random variables. Each parameter is described by a distribution rather than an individual value. Uncertainty in our parameter estimates is naturally channeled into any predictions we make. We saw two examples of *prior* and *likelihood* combinations that were *conjugate*, meaning that the *posterior* would be of the same form as the prior and could be computed analytically. Examples where we can justify the choice of a conjugate prior and likelihood combination are rare. In the remainder, we cannot compute the posterior and must resort to approximations. In this chapter, we will introduce three such approximation techniques.

4.1 NON-CONJUGATE MODELS

In the previous chapter we saw two models for which exact Bayesian inference was possible. In the first case, we were modelling the tossing of a coin and the combination of a beta prior and binomial likelihood meant that we could state that the posterior would also belong to the beta family. In the second example, a Gaussian prior coupled with a Gaussian likelihood resulted in a Gaussian posterior. The fact that we knew the form of the posterior meant that we didn't need to calculate the normalisation constant (the denominator in, for example, Equation 3.3). As long as we could find something proportional to the density of interest (i.e. proportional to a beta or a Gaussian), we could be certain that the normalisation would take care of itself. The beta-binomial and Gaussian-Gaussian combinations are not the only conjugate priorlikelihood pairs that we can use. Two other popular examples are the multinomial-Dirichlet and the gamma-Gaussian for discrete and continuous data, respectively.

For many models, it is not possible (or not justifiable from a modelling perspective) to pick a conjugate prior and likelihood, and we are forced to approximate. In this chapter, we will introduce three approximation techniques through a binary classification problem. Binary classification is a common problem within machine learning and one for which no conjugate prior and likelihood combination exists. The three techniques that we will look at are a point estimate, an approximate density, and sampling. All three are widely used within machine learning.

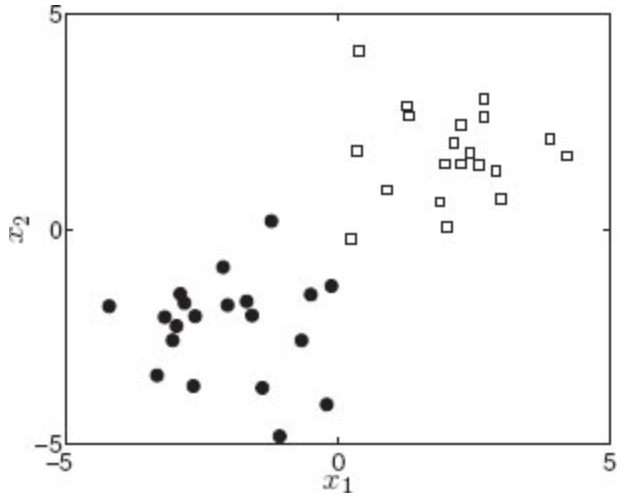


FIGURE 4.1 An example of a dataset with a binary response. Each object is defined by two attributes (x_1 and x_2) and a binary target, t = {0, 1}. Points with t = 0 are plotted as circles and those with t = 1 as squares.

4.2 BINARY RESPONSES

Figure 4.1 shows a dataset that looks a bit different from those we have seen so far. Each object is described by two attributes, x_1 and x_2, and has a binary response, t = {0, 1}. The objects are plotted with a symbol that depends on their response: if t = 0, the point is plotted as a circle, and, if t = 1, as a square. We will use this data to build a model that will enable us to predict the response (0 or 1; circle or square) for a new object. This task is known as classification – we want to be able to classify objects into one of a set of classes (in this case there are two classes). Classification is one of the major problems within machine learning, and we will introduce several other classification algorithms in Chapter 5.

4.2.1 A model for binary responses

We will work with the following vector and matrix representations of our data:

$$\mathbf{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_N^{\mathrm{T}} \end{bmatrix}.$$

Our model (with parameters **w**) will allow us to predict t_{new} for some new observation x_{new}.

Just as in our Olympics example in Section 3.8, we will need to compute the posterior density over the parameters of the model. According to Bayes' rule, this is given by

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{t}|\mathbf{X})} \tag{4.1}$$

where the marginal likelihood $p(\mathbf{t}|\mathbf{X})$ is given by

$$p(\mathbf{t}|\mathbf{X}) = \int p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w}.$$

Prior: We shall use a Gaussian density for the prior, $p(\mathbf{w})$. In particular, $p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. To be consistent, given that $p(\mathbf{w})$ depends on σ^2, we will denote the prior as $p(\mathrm{w}|\sigma^2)$. In previous chapters, the choice of a Gaussian density was often motivated by analytical convenience. Given that we are not going to be able to rely on conjugacy in this chapter, we are not restricted in our choice of prior density. However, our interest in this chapter is in the methods required to overcome nonconjugacy and for that, a Gaussian will suffice. Readers are recommended to try the methods introduced in this chapter with different forms of prior density, $p(\mathbf{w})$.

Likelihood: To make headway with the likelihood, $p(\mathbf{t}|\mathbf{X}, \mathbf{w})$, we start by assuming that the elements of t are conditionally independent (see Section 2.8.1), conditioned on $\mathbf{w}$:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w}\ ).$$

t_n is a binary variable indicating the class (0 or 1) of the nth object, $\mathbf{x}_n$. In the Gaussian Olympics example in the previous chapter, we treated t_n as a Gaussian random variable with mean $\mathbf{w}^{\mathsf{T}}\mathbf{x}_n$ and variance σ^2, but this is only appropriate for real-valued t_n. Instead, we can model t_n as a binary random variable – a single coin toss for each n. Rather than a mean and variance, this random variable is characterised by the probability that the class is 1 (the probability of belonging to class 0 is 1 minus the probability of belonging to class 1). To avoid confusion, we will denote this random variable T_n (to distinguish it from the actual instance, t_n, that we observe). Therefore, we can write each of the n likelihood terms as a probability:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} P(T_n = t_n|\mathbf{x}_n, \mathbf{w}). \tag{4.2}$$

This likelihood function will be high if the model assigns high probabilities for class 1 when we observe class 1 and high probabilities for class 0 when we observe class 0. It has a maximum value of 1 where all of the training points are predicted perfectly.

Our task is now to choose a function of $\mathbf{x}_n$ and $\mathbf{w}$, $f(\mathbf{x}_n; \mathbf{w})$, that produces a probability. A popular technique is to take a simple linear function (e.g. $f(\mathbf{x}_n; \mathbf{w}) = \mathbf{w}^{\mathsf{T}}\mathbf{x}_n$) and then pass the result through a second function that *squashes* its output to ensure it produces a valid probability. One such squashing function is the sigmoid function shown in Figure 4.2. As $\mathbf{w}^{\mathsf{T}}\mathbf{x}$ increases, the value converges to 1 and as it decreases, it converges to 0. The sigmoid function is defined as

$$P(T_n = 1|\mathbf{x}_n, \mathbf{w}) = \frac{1}{1+\exp(-\mathbf{w}^{\mathsf{T}}\mathbf{x}_n)}. \tag{4.3}$$

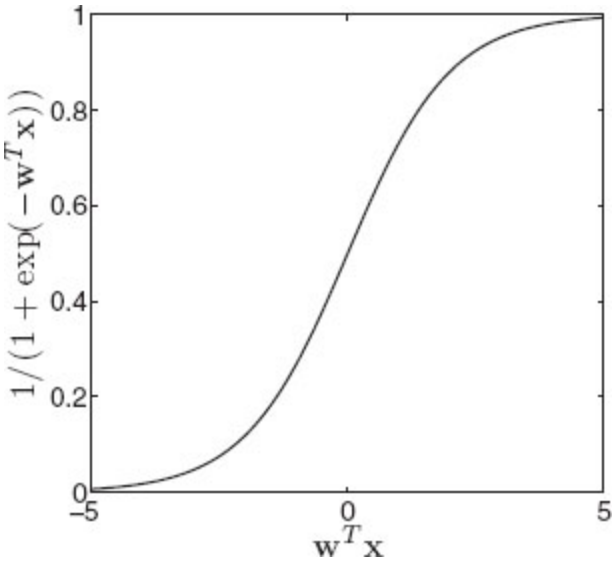


FIGURE 4.2 The sigmoid function that squashes a real value (e.g. $\mathbf{w}^{\mathrm{T}}\mathbf{x}$) to always be between 0 and 1.

This expression gives us the probability that T_n = 1. In our likelihood we require the probability of the actual observation, some of which will be zero. Because T_n can *only* take the value 0 or 1, we can easily compute $P(T_n = 0 | \mathbf{x}, \mathbf{w})$ using Equation 2.2:

$$\begin{aligned} P(T_n = 0|\mathbf{x}_n, \mathbf{w}) &= 1 - P(T_n = 1|\mathbf{x}_n, \mathbf{w}) \\ &= 1 - \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x}_n)} \\ &= \frac{\exp(-\mathbf{w}^T\mathbf{x}_n)}{1+\exp(-\mathbf{w}^T\mathbf{x}_n)}. \end{aligned} \tag{4.4}$$

We combine Equations 4.3 and 4.4 to produce a single expression for $P(T_n = t_n | \mathbf{x}_n, \mathbf{w})$ as follows:

$$P(T_n = t_n|\mathbf{x}_n\mathbf{w}) = P(T_n = 1|\mathbf{x}_n, \mathbf{w})^{t_n} P(T_n = 0|\mathbf{x}_n, \mathbf{w})^{1-t_n},$$

where the observed data (t_n) switches the relevant term on and the other off.

Substituting this into Equation 4.2 gives us the likelihood for all n training points:

$$\begin{aligned} p(\mathbf{t}|\mathbf{X}, \mathbf{w}) &= \prod_{n=1}^{N} P(T_n = 1|\mathbf{x}_n\mathbf{w})^{t_n} P(T_n = 0|\mathbf{x}_n, \mathbf{w})^{1-t_n} \\ &= \prod_{n=1}^{N} \left(\frac{1}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)}\right)^{t_n} \left(\frac{\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)}\right)^{1-t_n}. \end{aligned} \tag{4.5}$$

Posterior: This definition of the likelihood combined with the Gaussian prior we chose earlier are all we need, in theory, to work out the posterior density, $p(\mathbf{w} | \mathbf{X}, \mathbf{t}, \sigma^2)$. Once we have the posterior density, we can predict the response (class) of new objects by taking an expectation with respect to this density:

$$P(t_{\mathrm{new}} = 1|\mathbf{x}_{\mathrm{new}}, \mathbf{X}, \mathbf{t}) = \mathbf{E}_{p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2)} \left\{ \frac{1}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{new}})} \right\}.$$

In practice, this is not straightforward. The posterior is not of any standard form. To be able to evaluate it at a particular **w**, we would need to evaluate both the numerator and denominator of Equation 4.1. The numerator is fine – we could evaluate the Gaussian prior density at **w** and the likelihood that we've just defined and multiply the two values together. The denominator is the problem, as we cannot analytically perform the integration required to compute the marginal likelihood:

$$Z^{-1} = p(\mathbf{t}|\mathbf{X}, \sigma^2) = \int p(\mathbf{t}|\mathbf{X}, \mathbf{w}) p(\mathbf{w}|\sigma^2) d\mathbf{w}.$$

In other words, we have a function $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2) = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)$ which we know is proportional to the posterior, $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2) = Zg(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$, but we do not know the constant of proportionality, Z^{-1} (note that this constant is traditionally defined as Z_{-1} rather than Z). We are left with three options:

1. Find the single value of **w** that corresponds to the highest value of the posterior. As $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$ is proportional to the posterior, a maximum of $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$ will also correspond to a maximum of the posterior. Z^{-1} is not a function of **w**.
2. Approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with some other density that we can compute analytically.
3. Sample directly from the posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$, knowing only $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$.

The first option is not very Bayesian – we will have to make predictions for new objects based on a single value of **w** and not a density. It is, however, easy to do and this makes it a popular technique. The second option leaves us with a density that is easy to work with (we can choose any density we like) but if the chosen density is very different from the posterior, our model will not be very reliable. The final option allows us to sample from the posterior (and hence get good approximations to any expectations that we might require) but can be difficult.

These are the three options that are open to us in any problem where we cannot directly compute the posterior density. All three options have good and bad points and the choice of one over another will depend on the specifications (and computational limitations) of the problem at hand. We will now describe each in turn.

4.3 A POINT ESTIMATE – THE MAP SOLUTION

In the previous section we showed that, whilst we could not compute the posterior density $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$, we could compute something proportional to it, $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$. This is equal to the prior multiplied by the likelihood. The value of **w** that maximises $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$ will also correspond to the value at the maximum of the posterior. This will be the single most likely value of $\widehat{\mathbf{w}}$ (under the posterior) and is a sensible choice if we decide to use a point estimate. Chapter 2 was devoted to finding the value of $\widehat{\mathbf{w}}$ that maximised the likelihood. The idea here is very similar except now we are maximising the likelihood multiplied by the prior. This solution is the maximum a posteriori (MAP) estimate that we first saw in Section 3.8.4 and is common within machine learning.

Comment 4.1 – The Newton–Raphson method: The Newton–Raphson method (also known as the Newton method) is a general method for finding points where functions are equal to zero, i.e., finding points where the function $f(x) = 0$. Given a current estimate of the zero point, x_n, we update it by moving to the point where the tangent to the function at x_n passes through the x-axis. This point can be computed by approximating the gradient as a change in $f(x)$ divided by a change in x. Defining $\partial f(x)/\partial x$ as $f'(x)$,

$$\begin{aligned} f'(x_n) &= \frac{f(x_n)-0}{x_n-x_n+1} \\ (x_n - x_n + 1)f'(x_n) &= f(x_n) \\ x_n + 1 &= x_n - \frac{f(x_n)}{f'(x_n)}. \end{aligned}$$

The method can also be used to find minima and maxima, as these are simply points where the gradient passes through zero. Therefore, we simply replace $f(x)$ with its derivative $f'(x)$ and $f'(x)$ with its derivative $f''(x)$:

$$x_n + 1 = x_n - \frac{f'(x_n)}{f''(x_n)}.$$

This is readily extendable to functions of a vector – say $\mathbf{x}$. In this instance, $f'(x_n)$ is replaced by the vector of partial derivatives evaluated at $\mathbf{x}_n$ and $1/f''(x_n)$ is replaced by the inverse of the Hessian matrix (see Comment 2.6) – $\partial^2 f(\mathbf{x})/\partial\mathbf{x}\partial\mathbf{x}^{\mathrm{T}}$ – evaluated at $\mathbf{x} = \mathbf{x}_n$.

As with finding the maximum likelihood solution, it is easiest to find the value of $\mathbf{w}$ that maximises log $g(\mathbf{w}; \mathbf{X}, \mathbf{t})$ rather than $g(\mathbf{w}; \mathbf{X}, \mathbf{t})$:

$$\log\ g(\mathbf{w}; \mathbf{X}, \mathbf{t}) = \log\ p(\mathbf{t}|\mathbf{X}, \mathbf{w}) + \log\ p(\mathbf{w}|\sigma^2).$$

Unlike the maximum likelihood solution for the linear model, we cannot obtain an exact expression for $\mathbf{w}$ by differentiating this expression and equating it to zero. Instead, we can use any one of many optimisation algorithms that start with a guess for $\mathbf{w}$ and then keep updating it in such a way that $g(\mathbf{w}; \mathbf{X}, \mathbf{t})$ increases until a maximum is reached. The Newton–Raphson procedure (see Comment 4.1) is one such method that updates $\mathbf{w}$ using the following equation:

$$\mathbf{w}' = \mathbf{w} - \left(\frac{\partial^2 \log\ g(\mathbf{w};\mathbf{X},\mathbf{t})}{\partial\mathbf{w}\partial\mathbf{w}^{\mathrm{T}}}\right)^{-1} \frac{\partial \log\ g(\mathbf{w};\mathbf{X},\mathbf{t})}{\partial\mathbf{w}}. \tag{4.6}$$

The new version ($\mathbf{w}'$) of $\mathbf{w}$ is calculated by subtracting the inverse of the Hessian (see Comment 2.6) multiplied by the vector of partial derivatives. For any starting value of $\mathbf{w}$, this iterative procedure will update $\mathbf{w}$ until it reaches a point where the gradient is zero. To check that the point we have converged to corresponds to a maximum, we can check the Hessian to ensure that it is negative definite, just as we did for maximum likelihood in Section 2.8.3.

In order to compute the vector of first derivatives, we first expand our expression for log $g(\mathbf{w}; \mathbf{X}, \mathbf{t})$ using Equations 4.2 and 4.5:

$$\begin{aligned} \log\ g(\mathbf{w}; \mathbf{X}, \mathbf{t}) &= \sum_{n=1}^{N} \log P(T_n = t_n|\mathbf{x}_n, \mathbf{w}) + \log p(\mathbf{w}|\sigma^2) \\ &= \sum_{n=1}^{N} \log \left[\left(\frac{1}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)} \right)^{t_n} \left(\frac{\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)} \right)^{1-t_n} \right] + \log p(\mathbf{w}|\sigma^2). \end{aligned}$$

To stop this expression becoming too complicated, we will use the following shorthand:

$$P_n = P(T_n = 1|\mathbf{w}, \mathbf{x}_n) = \frac{1}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)}.$$

Therefore, assuming that **w** is *D*-dimensional, we have the following expression:

$$\begin{aligned}\log\ g(\mathbf{w};\mathbf{X},\mathbf{t}) &= \log p(\mathbf{w}|\sigma^2) + \sum_{n=1}^{N} \log P_n^{t_n} + \log\left(1-P_n\right)^{1-t_n} \\ &= -\tfrac{D}{2}\log 2\pi - D\log\sigma - \tfrac{1}{2\sigma^2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N} t_n \log P_n + (1-t_n)\log(1-P_n),\end{aligned}$$

where the first three terms are the log of the (Gaussian) prior. To find the vector of partial derivatives, we can use the chain rule (see Comment 4.2) to give an expression in terms of the partial derivatives of P_n:

$$\begin{aligned}\frac{\partial \log\ g(\mathbf{w};\mathbf{X},\mathbf{t})}{\partial \mathbf{w}} &= -\tfrac{1}{\sigma^2}\mathbf{w} + \sum_{n=1}^{N}\left(\frac{t_n}{P_n}\frac{\partial P_n}{\partial \mathbf{w}} + \frac{1-t_n}{1-P_n}\frac{\partial(1-P_n)}{\partial \mathbf{w}}\right) \\ &= -\tfrac{1}{\sigma^2}\mathbf{w} + \sum_{n=1}^{N}\left(\frac{t_n}{P_n}\frac{\partial P_n}{\partial \mathbf{w}} - \frac{1-t_n}{1-P_n}\frac{\partial P_n}{\partial \mathbf{w}}\right),\end{aligned} \tag{4.7}$$

where we have used the chain rule a second time to turn $\frac{\partial(1-P_n)}{\partial \mathbf{w}}$ into $-\frac{\partial P_n}{\partial \mathbf{w}}$:

$$\begin{aligned}\frac{\partial(1-P_n)}{\partial \mathbf{w}} &= \frac{\partial(1-P_n)}{\partial P_n}\frac{\partial P_n}{\partial \mathbf{w}} \\ &= \frac{\partial P_n}{\partial \mathbf{w}}.\end{aligned}$$

To calculate $\frac{\partial P_n}{\partial \mathbf{w}}$, we can use the chain rule once more:

$$\begin{aligned}\frac{\partial P_n}{\partial \mathbf{w}} &= \frac{\partial(1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n))^{-1}}{\partial(1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n))}\frac{\partial(1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n))}{\partial \mathbf{w}} \\ &= -\frac{1}{(1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n))^2}\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)(-\mathbf{x}_n) \\ &= \frac{\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)}{(1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n))^2}\mathbf{x}_n \\ &= \frac{1}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)}\frac{\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x})}\mathbf{x}_n \\ &= P_n(1-P_n)\mathbf{x}_n.\end{aligned} \tag{4.8}$$

Comment 4.2 – The chain rule: When taking partial derivatives, it is often convenient to use the chain rule. The chain rule states that

$$\frac{\partial f(g(\mathbf{w}))}{\partial \mathbf{w}} = \frac{\partial f(g(\mathbf{w}))}{\partial g(\mathbf{w})}\frac{\partial g(\mathbf{w})}{\partial \mathbf{w}}.$$

As an example, let

$$f(\mathbf{w}) = t_n \log P_n$$

where

$$P_n = \frac{1}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)}.$$

To compute $\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}}$, we can use the chain rule as follows:

$$\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial f(\mathbf{w})}{\partial P_n}\frac{\partial P_n}{\partial \mathbf{w}} = \frac{t_n}{P_n}\frac{\partial P_n}{\partial \mathbf{w}}.$$

Substituting Equation 4.8 into Equation 4.7 gives us the required vector of partial derivatives:

$$\begin{aligned}\frac{\partial \log\ g(\mathbf{w};\mathbf{X},\mathbf{t})}{\partial \mathbf{w}} &= -\frac{1}{\sigma^2}\mathbf{w} + \sum_{n=1}^{N}(\mathbf{x}_n t_n(1-P_n) - \mathbf{x}_n(1-t_n)P_n) \\ &= -\frac{1}{\sigma^2}\mathbf{w} + \sum_{n=1}^{N}\mathbf{x}_n(t_n - t_n P_n - P_n + t_n P_n) \\ &= -\frac{1}{\sigma^2}\mathbf{w} + \sum_{n=1}^{N}\mathbf{x}_n(t_n - P_n).\end{aligned} \tag{4.9}$$

To compute the Hessian matrix of second derivatives, we differentiate this again with respect to $\mathbf{w}^{\mathrm{T}}$. Noting that $\frac{\partial P_n}{\partial \mathbf{w}^{\mathrm{T}}} = \left(\frac{\partial P_n}{\partial \mathbf{w}}\right)^{\mathrm{T}}$, we obtain the following expression:

$$\begin{aligned}\frac{\partial^2 \log\ g(\mathbf{w};\mathbf{X},\mathbf{t})}{\partial \mathbf{w}\partial \mathbf{w}^{\mathrm{T}}} &= -\frac{1}{\sigma^2}\mathbf{I} - \sum_{n=1}^{N}\mathbf{x}_n\frac{\partial P_n}{\partial \mathbf{w}^{\mathrm{T}}} \\ &= -\frac{1}{\sigma^2}\mathbf{I} - \sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}P_n(1-P_n).\end{aligned} \tag{4.10}$$

One thing to notice from the Hessian is that, because $0 \leq P_n \leq 1$, it will be negative definite for any set of $\mathbf{x}_n$ and for any $\mathbf{w}$ (see Section 2.8.3). Therefore, there can only be one optimum and it must be a maximum. Whatever value of $\mathbf{w}$ the Newton–Raphson procedure converges to must correspond to the highest value of the posterior density. This is a consequence of the choice of prior and likelihood function, and changing either may result in a harder posterior density to optimise.

We now have everything we need to perform the Newton–Raphson procedure and find a potential optimal value of $\mathbf{w}$. Starting with $\mathbf{w} = [0, 0]^{\mathrm{T}}$ and setting $\sigma^2 = 10$, the procedure converges (the change in $\mathbf{w}$ becomes insignificant) after only nine iterations (MATLAB script: **logmap.m**). The evolution of the two components of $\mathbf{w}$ over this period can be seen in Figure 4.3. Following the previous chapters, we will call the value of $\mathbf{w}$ that corresponds to the maximum $\widehat{\mathbf{w}}$

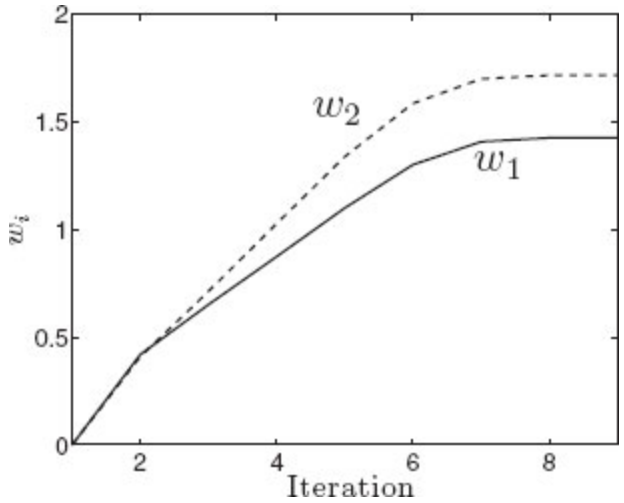


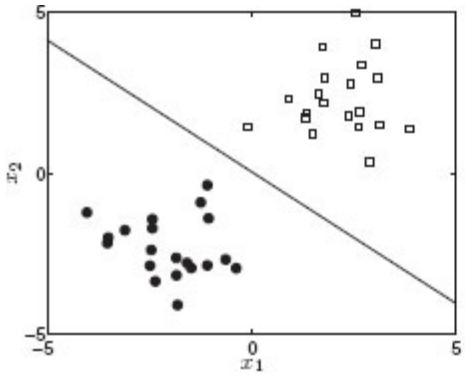
FIGURE 4.3 Evolution of the components of w throughout the Newton–Raphson procedure to find the w corresponding to the maximum of the posterior density.

Using $\widehat{\mathbf{w}}$, we can compute the probability that the response equals 1 for any **x**. In particular, if we observe $\mathbf{x}_{\text{new}}$, a new set of attributes, the probability that it should be given a response of 1 (it belongs to the square class) is given by

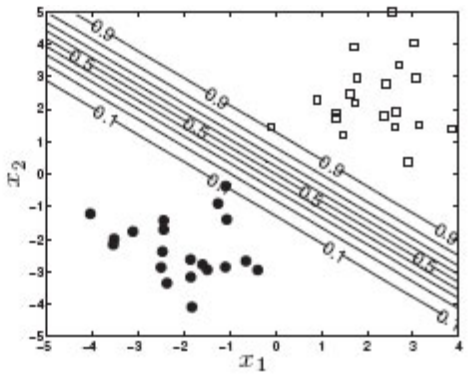
$$P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \widehat{\mathbf{w}}) = \frac{1}{1+\exp(-\widehat{\mathbf{w}}^{\mathrm{T}}\mathbf{x}_{\text{new}})}. \tag{4.11}$$

Given that there are two possible responses (or classes) for this new object, a sensible strategy might be to assign it to the square class (T_{new} = 1) if the probability is greater than 0.5 and to the circle class (T_{new} = 0) otherwise. In this case, the set of **x** values that correspond to $P(T = 1 | \mathbf{x}, \widehat{\mathbf{w}}) = 0.5$ will form a line that can be thought of as a **decision boundary** – points on one side of the line will belong to one class, and points on the other side to the other class. To plot the decision boundary, we make use of the fact that $P(T = 1 | \mathbf{x}, \widehat{\mathbf{w}}) = 0.5$ implies that $\widehat{\mathbf{w}}^{\mathrm{T}}\mathbf{x} = 0$ (see Exercise 4.5). If we expand this expression, we can obtain the decision boundary as a function of x_1 and x_2:

$$\begin{aligned} 0 &= \widehat{\mathbf{w}}^{\mathrm{T}}\mathbf{x} \\ &= \widehat{w_1}x_1 + \widehat{w_2}x_2 \\ \widehat{w_2}x_2 &= \widehat{-w_1}x_1 \\ x_2 &= -\frac{\widehat{w_1}x_1}{\widehat{w_2}}, \end{aligned}$$



(a) The data and the line where $P(T = 1|\mathbf{x}, \hat{\mathbf{w}}) = 0.5$. If we took 0.5 as a threshold, new points above the line would be classified as squares, below as circles.



(b) The contours show $P(T = 1|\mathbf{x}, \hat{\mathbf{w}})$, the probability that a new object should be classified as a square, as a function of $\mathbf{x}$.

FIGURE 4.4 Inferred function in the binary response example.

which is plotted in Figure 4.4(a). If we want to split the two classes with a straight line, this seems like quite a reasonable choice. In Figure 4.4(b) we plot contours of $P(T = 1|\mathbf{x}, \widehat{\mathbf{w}})$ as a function of $\mathbf{x}$ (MATLAB script: **logmap.m**). Close to the squares the probability is 1 (the squares are objects for which $t_n = 1$) and close to the circles it is 0. Between the two groups of data, the probability is around 0.5, reflecting the fact that objects here would be equidistant from both groups.

The outcome of this optimisation is that we have a model with which we can make predictions. The model is based on a point estimate, $\widehat{\mathbf{w}}$, of the parameters that we have obtained by finding the value of $\mathbf{w}$ that corresponds to a maximum of the posterior, $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$. This MAP solution is common in machine learning because it is reasonably easy to find $\widehat{\mathbf{w}}$ in this way. One could follow the steps described above for any prior and likelihood combination and find an optimum value. The optimisations will not always be as well behaved as this – in some problems, the posterior might have several maxima (and maybe even some minima). It would be difficult to know if the maximum we had found using Newton–Raphson was the **global optimum**.

In Chapter 3 we have already seen the advantage of maintaining a density over $\mathbf{w}$ rather than collapsing onto a point estimate. With this in mind, we will now move on to our second option when faced with a posterior we cannot compute exactly – finding a density that approximates $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$.

4.4 THE LAPLACE APPROXIMATION

There are various approximation methods used within machine learning to replace tricky posterior densities with approximations that are easier to handle. The most popular is the Laplace approximation.[1] The idea is to approximate the density of interest with a Gaussian. Given the ease with which we can manipulate Gaussians, this seems to be a sensible choice – the expectations required to make predictions are likely to be easy to calculate given a Gaussian posterior. However, we should always bear in mind that our predictions will then only be as good as our approximation. If our true posterior is not very Gaussian, our predictions will be easy to compute but not very useful.

The Gaussian density is defined by its mean and (co)variance. Using a Gaussian to approximate another density amounts to choosing suitable values for these parameters. To motivate the choices of parameters made by the Laplace approximation, imagine that, rather than having two parameters, our model has only one – w – and that we know $\widehat{w}$ – the value corresponding to the highest value of the posterior. Our first step is to approximate $\log g(w; \mathbf{X}, \mathbf{t}, \sigma^2)$ using a Taylor expansion (see Comment 4.3) around the maximum, $\widehat{w}$:

$$\log\ g(w;X,\mathbf{t},\sigma^2) \approx \log\ g(\widehat{w};\mathbf{X},\mathbf{t},\sigma^2) + \frac{\partial \log\ g(w;\mathbf{X},\mathbf{t},\sigma^2)}{\partial w}\Big|_{\widehat{w}} \frac{(w-\widehat{w})}{1!} + \frac{\partial^2 \log\ g(w;\mathbf{X},\mathbf{t},\sigma^2)}{\partial w^2}\Big|_{\widehat{w}} \frac{(w-\widehat{w})^2}{2!} + \ldots$$

The second term is the first derivative (i.e. the gradient) evaluated at the maximum point and must therefore be zero. Discarding this, and ignoring terms of third-order and above, we are left with the following expression:

$$\log\ g(w;\mathbf{X},\mathbf{t},\sigma^2) \approx \log\ g(\widehat{w};\mathbf{X},\mathbf{t},\sigma^2) - \frac{v}{2}(w-\widehat{w})^2, \tag{4.12}$$

where v is the negative of the second derivative of $\log g(w; \mathbf{X}, \mathbf{t}, \sigma^2)$ evaluated at $w = \widehat{\boldsymbol{w}}$:

$$v = -\frac{\partial^2 \log g(w;\mathbf{X},\mathbf{t},\sigma^2)}{\partial w^2}\Big|_{\widehat{w}}.$$

Now, the Gaussian density is defined as

$$\frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2\sigma^2}(w-\mu)^2\right\},$$

the log of which is equal to

$$\log(K) - \frac{1}{2\sigma^2}(w-\mu)^2,$$

where K is the normalising constant. This looks very similar to Equation 4.12 with $\mu = \widehat{\boldsymbol{w}}$ and $\sigma^2 = 1/v$. This is the Laplace approximation – we approximate the posterior with a Gaussian that has its mean at the posterior **mode** ($\widehat{\boldsymbol{w}}$) and has variance inversely proportional to the curvature of the posterior (its second derivative) at its mode.

Comment 4.3 – Taylor expansions: The Taylor expansion is a way of approximating a function. The approximation is always made 'about' some value – the approximation will tend to diverge from the true function as we move away from that value. The definition of the Taylor series of $f(w)$ about $\widehat{\boldsymbol{w}}$ is

$$f(w) = \sum_{n=0}^{\infty} \frac{(w-\widehat{w})^n}{n!} \frac{\partial^n f(w)}{\partial w^n}\Big|_{\widehat{w}}$$

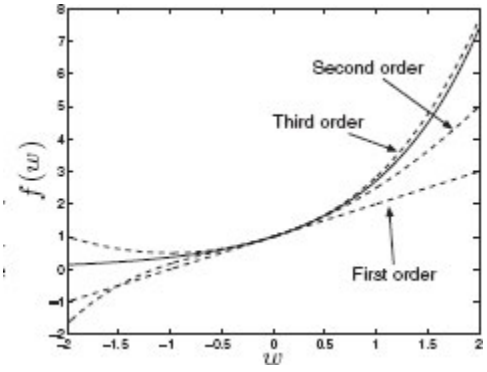
where $\frac{\partial^n f(w)}{\partial w^n}\Big|_{\widehat{w}}$ is the nth derivative of $f(w)$ with respect to w, evaluated at $\widehat{\boldsymbol{w}}$. When $n = 0$, this derivative is simply the function $f(w)$. If we only compute a finite number of terms, we will have an approximation to the function. A first-order approximation would just include terms $n = 0$ and $n = 1$ – an nth-order approximation includes all terms up to and including term n. For example, we can approximate $f(w) = \exp(w)$ at $\widehat{\boldsymbol{w}} = 0$:

$$\exp(w) = \exp(\widehat{w}) + \frac{w}{1!}\exp(\widehat{w}) + \frac{w^2}{2!}\exp(\widehat{w}) + \ldots$$

Now, $\exp(\widehat{\boldsymbol{w}}) = 1$, so

$$\exp(w) = 1 + \frac{w}{1!} + \frac{w^2}{2!} + \frac{w^3}{3!} + \ldots$$

The approximation will get better and better as we add more and more terms. This can be seen in the figure on the right.



This idea is easily extended to multivariate densities. In particular, the Laplace approximation to our true posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ is

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2) \approx \mathcal{N}(\mu, \mathbf{\Sigma}),$$

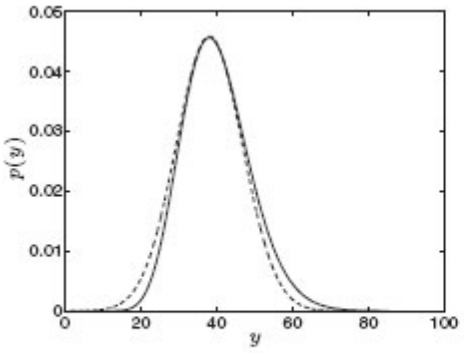
where μ is set to $\widehat{\boldsymbol{w}}$, and $\mathbf{\Sigma}$ is the negative of the inverse Hessian:

$$\mu = \widehat{\mathbf{w}}, \quad \mathbf{\Sigma}^{-1} = -\left(\frac{\partial^2 \log\, g(\mathbf{w};\mathbf{X},\mathbf{t})}{\partial \mathbf{w} \partial \mathbf{w}^{\mathrm{T}}}\right)\Big|_{\widehat{\mathbf{w}}}. \tag{4.13}$$

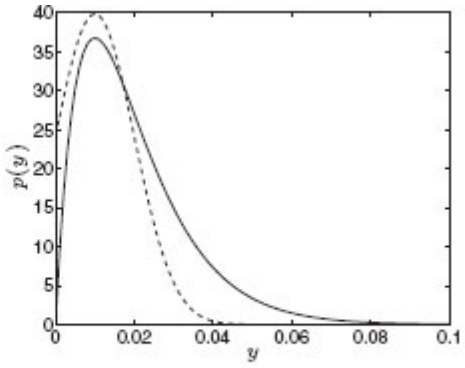
4.4.1 Laplace approximation example: Approximating a gamma density

Before we look at what this approximation looks like in the binary response example, it is useful to look at an example where we know the true density (see also Exercises 4.1, 4.2 and 4.3) (MATLAB script: **lapexample.m**). This will allow us to see how good or bad the approximation is. The following is the gamma density for a random variable Y:

$$p(y|\alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} y^{\alpha - |1} \exp\Big\{-\beta y\Big\}. \tag{4.14}$$



(a) $p(y|\alpha, \beta)$ (solid line) and approximating Gaussian (dashed line) for $\alpha = 20$, $\beta = 0.5$.



(b) $p(y|\alpha, \beta)$ (solid line) and approximating Gaussian (dashed line) for $\alpha = 2$, $\beta = 100$.

FIGURE 4.5 Examples of the Laplace approximation to the gamma density function given in Equation 4.14.

We will investigate how good the Laplace approximation is to this density. The gamma density has an analytic expression for its mode which means we do not need to go through an optimisation procedure similar to that in the last section. The mode, $\hat{y}$, is defined as

$$\hat{y} = \frac{\alpha - 1}{\beta}.$$

The Laplace approximation to $p(y|\alpha, \beta)$ takes the form of a Gaussian:

$$p(y|\alpha, \beta) \approx \mathcal{N}(\mu, \sigma^2).$$

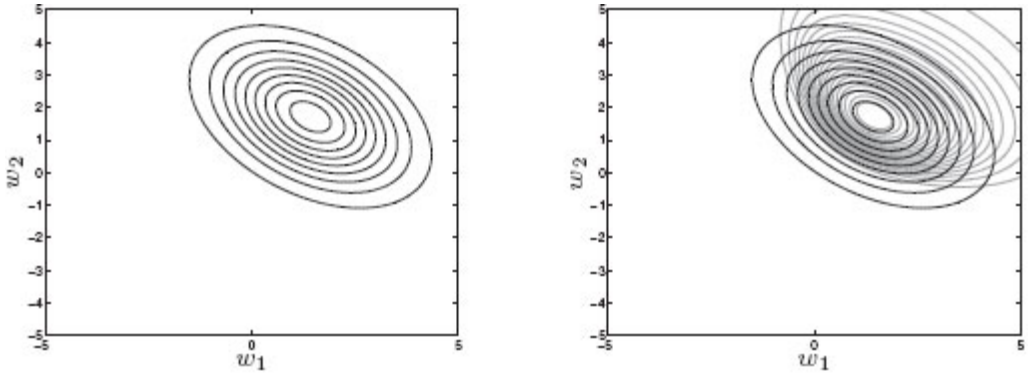
The mean μ will be equal to the mode of $p(y|\alpha, \beta)$, which we've already defined. To find the variance, σ^2, of the approximating Gaussian, we need to find the second derivative of $\log p(y|\alpha, \beta)$ with respect to y. This is computed as follows:

$$\begin{aligned}
\log p(y|\alpha, \beta) &= \alpha \log \beta - \log(\Gamma(\alpha)) + (\alpha - 1) \log y - \beta y \\
\frac{\partial \log p(y|\alpha,\beta)}{\partial y} &= \frac{\alpha - 1}{y} - \beta \\
\frac{\partial^2 \log p(y|\alpha,\beta)}{\partial y^2} &= -\frac{\alpha - 1}{y^2}.
\end{aligned}$$

σ^2 will be equal to the negative inverse of this quantity evaluated at $y = \hat{y}$. In particular

$$\sigma^2 = \frac{\hat{y}^2}{\alpha - 1} = \frac{\alpha - 1}{\beta^2}.$$

In Figure 4.5 we can see two examples of $p(y|\alpha, \beta)$ and the corresponding Laplace approximation. In the first, $p(y|\alpha, \beta)$ looks rather like a Gaussian and the approximation is pretty good. In the second, $p(y|\alpha, \beta)$ does not look very much like a Gaussian and the approximation is not accurate. In both cases the approximation gets worse as we move away from the mode. This is because the approximation is based on the characteristics of the function *at* the mode. We will see this property again as we return to the binary response model.



(a) Laplace approximation to the posterior.

(b) Laplace approximation to the posterior and the true unnormalised posterior (lighter lines).

FIGURE 4.6 The Laplace approximation for the binary problem.

4.4.2 Laplace approximation for the binary response model

Returning to our binary response model, we had to compute both the mode, $\widehat{\mathbf{w}}$ and the Hessian for the Newton–Raphson procedure. We therefore already have everything we need for the Laplace

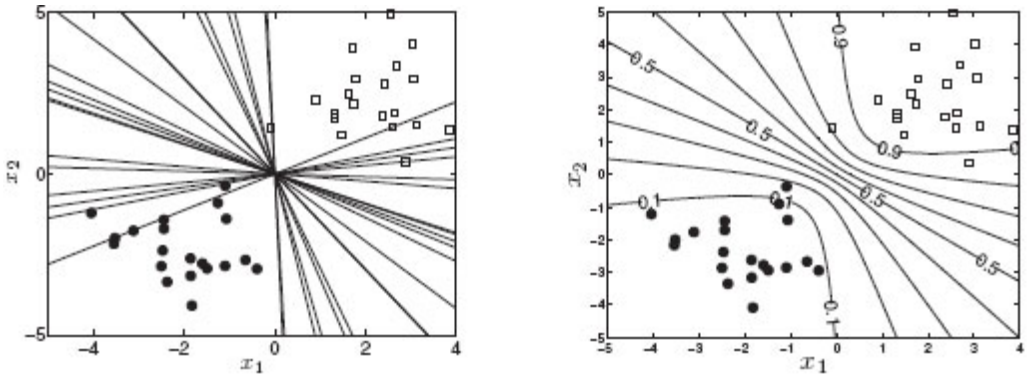
approximation to the posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$. In Figure 4.6(a) we can see the approximate posterior and in Figure 4.6(b) we can see the same approximation on top of $g(\mathbf{w}; \mathbf{X}, \mathbf{t})$, the unnormalised posterior. As for the gamma example in the previous section, the shape of the approximation is pretty good around the mode but diverges considerably from the true posterior as we move away from the mode. This is to be expected – the Laplace approximation only matches the shape (curvature) at the mode. We can also sample values of $\mathbf{w}$ from the approximate posterior and look at the decision boundaries that they correspond to. Twenty such boundaries are plotted in Figure 4.7(a). There appears to be a lot of variability in these boundaries, although all of them seem to split the classes reasonably well.

The final step is to use the approximate posterior to compute predictions. We now have a density over $\mathbf{w}$ rather than a single value and we know, from Chapter 3, that we compute a prediction by averaging over this density. In particular, we should be calculating the expected value of $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})$ with respect to the approximate posterior over $\mathbf{w}$ (which we've denoted as $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$):

$$P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \sigma^2) = \mathbf{E}_{\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\{P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w})\}.$$

Unfortunately, we cannot compute the integral over $\mathbf{w}$ required in this expectation. This might suggest that our choice of approximation was not sensible – we still cannot make predictions. However, we can easily sample from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and so (see Equation 2.23) we can approximate the expectation with

$$P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \sigma^2) = \frac{1}{N_s} \sum_{s=1}^{N_s} \frac{1}{1 + \exp(-\mathbf{w}_s^{\mathrm{T}} \mathbf{x}_{\text{new}})}, \tag{4.15}$$



(a) Twenty decision boundaries corresponding to instances of w sampled from the Laplace approximation to the posterior.

(b) Contours of the probability of belonging to class 1 computed with a sample based approximation to $\mathbf{E}_{\mathcal{N}(\mu,\Sigma)}\{P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w})\}$.

FIGURE 4.7 Decision boundaries sampled from the Laplace approximation and the predictive probability contours.

where $\mathbf{w}_s$ is the sth of N_s samples drawn from the approximate posterior. Using $N_s = 1000$, the contours of $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \sigma^2)$ can be seen in Figure 4.7(b) (MATLAB script: **loglap.m**). Compare this with Figure 4.4(b). There is a big difference – the contours are no longer straight lines. Averaging over the posterior density for $\mathbf{w}$ has had the effect of smudging the decision boundaries. The probabilities are now closer to 0.5 in all areas except those very close to the data objects. The model based on the point estimate, shown in Figure 4.4(b), could be said to be overconfident – take $x_1 = -3$, $x_2 = 5$ as an example. According to the predictions produced by the point estimate (Figure 4.4(b)), an object with these

attributes would have a probability of approximately 1 of being a square despite the fact that it is quite distant from the other square objects. Compare this with the probability of approximately 0.6 given by the expectation with respect to the Laplace approximation to the posterior (Figure 4.7(b)). This value seems much more reasonable. Another way to understand the uncertainty that should be present in areas like this is to look at Figure 4.7(a) – there is very large variability in the possible decision boundaries at $x_1 = -3$, $x_2 = 5$. Some of these boundaries would classify this object as a square, some as a circle – the probability that it is a square, given the data that we have seen, is not 1.

In this section we have seen again that we should be wary of using point estimates. The Laplace approximation shown here can be used to approximate any density (over real-valued random variables) for which we can find the mode and compute the second derivative. The approach assumes that the posterior can be reasonably approximated by a Gaussian, something that is not always the case (see Figure 4.5). In our binary response model, the approximation did not allow us to compute the expectation necessary for making predictions exactly. However, the ease with which we can sample from a Gaussian meant that it was straightforward to obtain a sample-based approximation to the expectation. In the next section, we will extend this idea through the introduction of a technique that will enable us to sample directly from $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ despite the fact that we cannot compute the normalisation constant. The ability to generate these samples will allow us to use a sample-based approximation to the expectation without having to approximate the posterior.

4.5 SAMPLING TECHNIQUES

The Laplace approximation in the previous section provided us with a method for approximating the posterior density $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$. Our interest in the posterior density is primarily to allow us to take all the uncertainty in $\mathbf{w}$ into account when making predictions. We do this by averaging over all potential values of $\mathbf{w}$ through the following expectation:

$$P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \sigma^2) = \mathbf{E}_{p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2)}\{P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w})\}.$$

Even substituting our approximation to the posterior into this expression, we could not analytically compute the integral required in this expectation. Fortunately, it was easy to sample from the Gaussian approximation, enabling us to use the sample based approximation given in Equation 4.15. In this instance, the benefit of making the approximation was that it enabled us to easily generate samples. In this section, we will look at a technique that enables us to cut out the approximation step and sample directly from the posterior. A set of samples from the true posterior generated in this way could be substituted directly into Equation 4.15 to compute the desired predictive probability, $P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \sigma^2)$. We're going to introduce a popular sampling technique known as the **Metropolis–Hastings** algorithm. However, before we go into this, it is perhaps useful to get more comfortable with the idea of sampling through a less abstract example.

4.5.1 Playing darts

In the game of darts, players take turns to throw three darts at a board like that shown in Figure 4.8. The darts are sharp and embed themselves into the board. The player receives a certain number of points for each dart, depending on where the dart lands. The scores from the three darts are added together and subtracted from the player's current total. Each player starts the game with the same total (normally 501) and the winner is the player who gets to zero first. The majority of the board is split into 20 segments and if the dart lands in the white parts of these segments, the score is equal to the number shown around the edge. If the dart lands in one of the shaded areas, the score is either double (lighter, outer shaded area) or triple (darker, inner area) the segment score. The circle in the centre of the board is known as the bull's-eye (50 points) and the circle around this as the bull (25 points). There is one slight complication to

the rules – the player must get to zero with a double. So, for example, if a player currently has a total of 40, they could win by throwing a double 20 (the lightly shaded area just below the '20' label) or a single 20 (anywhere in the white bits of the '20' segment) followed by a double 10, etc. We will assume that the player does indeed need to score 40 to win, and has only one dart left with which to