Machine Learning Module

Week 4

Lecture Notes 7 & 8

Probabilistic Classification Methods

Mark Girolami
girolami@dcs.gla.ac.uk
Department of Computing Science
University of Glasgow

February 3, 2006

1 Classification

A large class of problems which Machine Learning techniques are applied to are classification problems and in this section we will now look at a number of classification methods which are available to us.

As a simple example lets try and build a classifier which will predict whether a person is male or female based on their measured height alone.

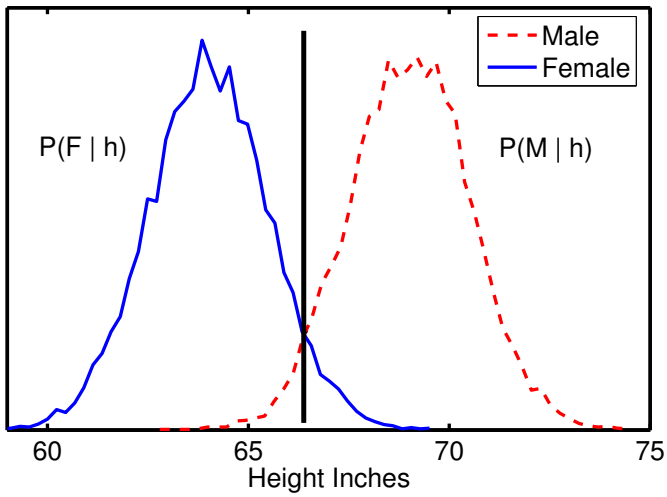

Figure 1: The distributions of measured height for both males and females in a population.

1.1 Class Priors

The class variable C will take on two values so we can encode male by the value 1 and female by the value 0. Now within the general population there is an approximate equal number of male and females (lets just assume that there is in any case for the time being). In that case the probability of class male occurring will be defined simply as $P(C = 1)$ and the probability of class female occuring will be $P(C = 0)$. Now these probabilities are set **prior** to making any measurements and hence are called the **prior probabilities** of class membership.

If these are well balanced i.e. $P(C = 0) = P(C = 1) = 0.5$ then it is equally likely to observe either class. However in applications such as

medical diagnostics or intrusion detection the prior probabilities of one class e.g. network intrusion or cancer are much smaller than the other e.g. normal traffic or not cancer. In this case then we can make a prediction before seeing any data that is more likely to be correct based on the prior probabilities alone.

1.2 Class Conditional Likelihood

Now we have an individual, randomly selected from the population, and we make a measurement of their height. Now there will be a natural distribution of the height of males and females, so in other words there will be a **class conditional distribution** of the measured features, in this case height. We can write these class conditional distributions as $p(h|C = 1)$ and $p(h|C = 0)$ form male and female classes respectively.

1.3 Class Posterior

Now from Bayes rule which we met last week we can obtain the posterior probability of class membership by noting that

$$P(h, C = 1) = p(h|C = 1)P(C = 1) = P(C = 1|h)p(h) \tag{1}$$

and so

$$P(C = 1|h) = \frac{p(h|C = 1)P(C = 1)}{p(h)} \tag{2}$$

and the marginal likelihood of our measurement, $p(h)$, is the probability of measuring a height h irrespective of the class and so

$$p(h) = p(h|C = 1)P(C = 1) + p(h|C = 0)P(C = 0) \tag{3}$$

which means that the class posteriors will also sum to one, $P(C = 1|h) + P(C = 1|h) = 1$.

1.4 Discriminant Functions

From Figure (1) we can see the empirical distributions of height for both males and females. The first thing to notice is that there is a distinct difference in the location of the distributions and they can be separated to a large extent (males are typically taller than females). However there is a region

where the two distributions overlap and it is here that classification errors can be made. The region of intersection where $P(C = 1|h) = P(C = 0|h)$ is important as it defines our decision boundary. If we make a measurement of 69 inches then we can see that $P(C = 1|h) > P(C = 0|h)$ and whilst there is some probability that we have measured a rather tall female, to minimise the unavoidable errors that will be made then our decision should be based on the largest posterior probability.

We can then define a **discriminant function** based on our posterior probabilities one such function could be the ratio of posterior probabilities for both classes. If we take the logarithm of this ratio then the general discriminant function

$$f(h) = \log \frac{P(C = 1|h)}{P(C = 0|h)} \tag{4}$$

would define the rules that h would be assigned to $C = 1$ (male) if $f(h) > 0$ and if $f(h) < 0$ the assignment would be to $C = 0$ (female).

1.5 Discriminative & Generative Classifiers

There are two ways in which we can define our discriminant function. In the first case we can explicitly model our discriminant function using for example a linear or nonlinear model. This is often referred to as the discriminative approach to defining a classifier as all effort is placed on defining the overall discriminant function with no consideration for the class-conditional densities which form the discriminant.

The second way is to focus on estimating the class-condition densities (distributions if the features are discrete) $p(h|C = 1)$ and $p(h|C = 0)$ and then use these estimates to define our posterior probabilities and hence our discriminant function. As the class-conditional densities define the statistical process which **generates** the features we measure then this approach is often referred to as the generative approach. We will introduce one example of both approaches starting with the discriminative approach.

2 Discriminative Approaches to Classification

As we have just considered the Bayesian formalism for linear models we will straightaway look at a Bayesian approach to classification using linear models. We will use the more general notation of $\mathbf{x} = [x_1, \cdots, x_D]^\mathsf{T}$ representing the D-dimensional vector containing each of D features available for classification purposes.

$$\log \frac{P(C=1|\mathbf{x})}{P(C=0|\mathbf{x})}$$

Now as the ratio of the probabilities of class membership $P(C=1|\mathbf{x})$ & $P(C=0|\mathbf{x})$ lies on the positive real line i.e. $[0+\infty)$ then the log-likelihood ratio will cover the whole of the real line i.e. take values between $-\infty$ and $+\infty$. As such we can model this ratio using a linear-model, where now we employ an explicit and general basis expansion of the input such that $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}, \cdots, \phi_M(\mathbf{x})]^\mathsf{T}$, and each $\phi_m(\mathbf{x})$ defines the m'th basis function applied to the data vector $\mathbf{x}$. We have already met simple polynomial basis functions previously, now we are being a little more general with our notation. Back to the log-likelihood ratio and our linear model of it

$$\log \frac{P(C=1|\mathbf{x})}{P(C=0|\mathbf{x})} = \mathbf{w}^\mathsf{T}\phi(\mathbf{x})$$

Now as $P(C=1|\mathbf{x}) + P(C=0|\mathbf{x}) = 1$ then a tiny little bit of algebra[1] shows that

$$\begin{aligned} \frac{P(C=1|\mathbf{x})}{1-P(C=1|\mathbf{x})} = \exp\left(\mathbf{w}^\mathsf{T}\phi(\mathbf{x})\right) \Rightarrow P(C=1|\mathbf{x}) &= \frac{1}{1+\exp\left(-\mathbf{w}^\mathsf{T}\phi(\mathbf{x})\right)} \\ &= \frac{\exp\left(\mathbf{w}^\mathsf{T}\phi(\mathbf{x})\right)}{1+\exp\left(\mathbf{w}^\mathsf{T}\phi(\mathbf{x})\right)} \end{aligned}$$

The likelihood for each data point (input-output pair) $(\mathbf{x}_n, t_n)$ will simply be the posterior probability $P(C = t_n|\mathbf{x}_n)$. This is a Logistic Regression model where the logistic function defines the posterior probability of class membership.

[1] Spoil yourself and work through this to convince yourself.

2.1 Bayesian Logistic Regression

Now we can write the likelihood component for each n as

$$\begin{aligned} P(C = t_n|\mathbf{x}_n, \mathbf{w}) &= P(C = 1|\mathbf{x}_n, \mathbf{w})^{t_n} \times (1 - P(C = 1|\mathbf{x}_n, \mathbf{w}))^{1-t_n} \\ &= \left[\frac{1}{1 + \exp\left(-\mathbf{w}^\mathsf{T}\phi(\mathbf{x}_n)\right)}\right]^{t_n} \left[\frac{1}{1 + \exp\left(\mathbf{w}^\mathsf{T}\phi(\mathbf{x}_n)\right)}\right]^{1-t_n} \\ &= \frac{\exp\left(\mathbf{w}^\mathsf{T}\phi(\mathbf{x}_n)\right)^{t_n}}{1 + \exp\left(\mathbf{w}^\mathsf{T}\phi(\mathbf{x}_n)\right)} \end{aligned}$$

Let us be bold and take a Bayesian viewpoint straightaway (you know it makes sense!) so we will place a Gaussian prior on our coefficients such that $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$ and we assume that each t_n is sampled i.i.d (remember this from last week?) in which case our likelihood will be

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

Now that we are all good Bayesians we immediately want to define the posterior over the parameters and so we need the joint-likelihood formed by the likelihood and the prior

$$\begin{aligned} p(\mathbf{t}, \mathbf{w}|\mathbf{X}, \alpha) &= p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\alpha) \\ &= \prod_{n=1}^{N} \frac{\exp\left(\mathbf{w}^\mathsf{T}\phi(\mathbf{x}_n)\right)^{t_n}}{1 + \exp\left(\mathbf{w}^\mathsf{T}\phi(\mathbf{x}_n)\right)} \mathcal{N}_\mathbf{w}(\mathbf{0}, \alpha^{-1}\mathbf{I}) \end{aligned}$$

Now of course we want our posterior, however, this is the point where I tell you about the fly in the ointment. To obtain our posterior we require the following

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \alpha) = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\alpha)\frac{1}{p(\mathbf{t}|\mathbf{X}, \alpha)}$$

where the marginal likelihood

$$
\begin{aligned}
p(\mathbf{t}|\mathbf{X},\alpha) &= \int p(\mathbf{t}|\mathbf{X},\mathbf{w})p(\mathbf{w}|\alpha)d\mathbf{w} \\
&= \int \prod_{n=1}^{N} \frac{\exp\left(\mathbf{w}^{\mathsf{T}}\phi(\mathbf{x}_n)\right)^{t_n}}{1+\exp\left(\mathbf{w}^{\mathsf{T}}\phi(\mathbf{x}_n)\right)} \mathcal{N}_{\mathbf{w}}(\mathbf{0},\alpha^{-1}\mathbf{I})d\mathbf{w}
\end{aligned}
$$

So here is the bad news, the above multi-dimensional integral cannot be computed analytically. Unlike the really nice regression problem where a fully analytic expression for the posterior was available in the classification setting we run into some small degree of difficulty and a number of avenues are open to us in order to make progress.

The first thing that we can do is solve the above integral numerically or go the whole hog and simulate samples from the full posterior and use these samples to compute any posterior expectations we require. This is the basis of Monte Carlo methods and for now we shall leave this method to the side.

The second thing that we can do is make an approximation of the posterior which will be analytically convenient, and this is what we shall now do.

2.2 Laplace Approximation

It can be shown that for large data samples i.e. $N \to \infty$ where N is much larger than the number of parameters, in our case the dimension of $\mathbf{w}$, then the parameter posterior distribution is approximately multivariate Gaussian with a mean value equal to the parameter values which yield the maximum of the posterior distribution and has a covariance matrix which captures the curvature of the posterior at the maximum value and is defined as the negative inverse for the matrix of partial derivatives computed at the maximum value (this should be familiar to you from our presentation of the maximum likelihood method in the previous lecture). In other words if we define the parameters at the maximum of the posterior as $\mathbf{w}_{MAP}$ and the covariance of the approximation as $\mathbf{C}$, where

$$
\mathbf{C} = -\left(\frac{\partial^2}{\partial\mathbf{w}\partial\mathbf{w}^{\mathsf{T}}}\log p(\mathbf{t},\mathbf{w}|\mathbf{X},\alpha)\right)^{-1}
$$

where the right-hand side is computed at the *MAP* value $\mathbf{w}_{MAP}$ in which case we can write

$$p(\mathbf{w}|\mathbf{t},\mathbf{X},\alpha) = p(\mathbf{t}|\mathbf{X},\mathbf{w})p(\mathbf{w}|\alpha)\frac{1}{p(\mathbf{t}|\mathbf{X},\alpha)} \approx \mathcal{N}_{\mathbf{w}}(\mathbf{w}_{MAP},\mathbf{C})$$

Now this means that we need to somehow estimate the *Maximum a Posteriori* parameter value as well as compute the curvature of the posterior at that point. Just note that we need to find the parameter values which maximise the posterior and we can do this by maximising the logarithm of the joint likelihood as the normalising term (the marginal) does not depend on the parameters. So as before let us write out the logarithm of the joint likelihood which follows as

$$\begin{aligned}\mathcal{L} = \log p(\mathbf{t},\mathbf{w}|\mathbf{X},\alpha) = \sum_{n=1}^{N} t_n \mathbf{w}^{\mathsf{T}}\phi(\mathbf{x}_n) &- \log\left(1+\exp\left(\mathbf{w}^{\mathsf{T}}\phi(\mathbf{x}_n)\right)\right) \\ &- \frac{1}{\alpha}\mathbf{w}^{\mathsf{T}}\mathbf{w} - \frac{D}{2}\log(2\pi\alpha^2)\end{aligned}$$

this is clearly not as nice an expression as we had for the linear regression models we have already met. Now let us take first and second derivatives with respect to all the parameter values $\mathbf{w}$.

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{w}} &= \sum_{n=1}^{N} t_n\phi(\mathbf{x}_n) - P(C=1|\mathbf{x}_n)\phi(\mathbf{x}_n) - \frac{1}{\alpha}\mathbf{w} \\ &= \mathbf{\Phi}^{\mathsf{T}}\mathbf{t} - \mathbf{\Phi}^{\mathsf{T}}\mathbf{p} - \frac{1}{\alpha}\mathbf{w}\end{aligned}$$

where the $N\times 1$ vector of class-membership probabilities is defined as $\mathbf{p} = [P(C=1|\mathbf{x}_1),\cdots,P(C=1|\mathbf{x}_N)]^{\mathsf{T}}$ and the $N\times M$ matrix $\mathbf{\Phi}$ is defined as

$$\mathbf{\Phi} = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \cdots & \phi_M(\mathbf{x}_1) \\ \vdots & \phi_m(\mathbf{x}_n) & \vdots \\ \phi_1(\mathbf{x}_N) & \cdots & \phi_M(\mathbf{x}_N) \end{bmatrix}$$

The second-derivatives follows as before

$$\begin{aligned}\frac{\partial^2\mathcal{L}}{\partial\mathbf{w}\partial\mathbf{w}^{\mathsf{T}}} &= -\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{\mathsf{T}}P(C=1|\mathbf{x}_n)\left(1-P(C=1|\mathbf{x}_n)\right) - \frac{1}{\alpha}\mathbf{I} \\ &= -\mathbf{\Phi}^{\mathsf{T}}\mathbf{V}\mathbf{\Phi} - \frac{1}{\alpha}\mathbf{I}\end{aligned}$$

where $\mathbf{V}$ is an $N \times N$ dimensional diagonal matrix defined as

$$\begin{bmatrix} v_{11} & 0 & \cdots & 0 \\ 0 & v_{22} & \cdots & 0 \\ \vdots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & v_{NN} \end{bmatrix}$$

where each $v_{nn} = P(C = 1|\mathbf{x}_n)(1 - P(C = 1|\mathbf{x}_n))$. **You are strongly encouraged to work through the derivatives manually and convince yourself of your ability to derive the above results**.

Now then we can define the covariance matrix of the *approximate* posterior as

$$\mathbf{C} = \left(\mathbf{\Phi}^{\mathsf{T}}\mathbf{V}\mathbf{\Phi} + \frac{1}{\alpha}\mathbf{I}\right)^{-1}$$

You should notice the similarity of this and the covariance of the posterior under a linear regression model (refer to last weeks notes).

The *MAP* value for the parameters does not follow in the nice closed form by setting the gradients to zero and solving for $\mathbf{w}$ as in the standard linear regression model as each element of the vector $\mathbf{p}$ i.e. $P(C = 1|\mathbf{x}_n)$ is itself a nonlinear function of $\mathbf{w}$. We now need to resort to optimisation techniques.

2.3 Newton Optimisation Routine

We need to find the parameter values $\mathbf{w}_{MAP}$ which will yield the maximum is to make moves in parameter space which will yield the largest change in the criterion to be maximised, in this case the joint likelihood. This can be achieved by making changes to the parameters in the direction of steepest ascent, so in other words follow the gradient. So the change in the parameters $\Delta\mathbf{w}$ would be proportional to the gradient computed at that point in parameter space.

$$\Delta\mathbf{w} = \delta\frac{\partial\mathcal{L}}{\partial\mathbf{w}}$$

where δ is the *step-size* which is taken when moving from $\mathbf{w}_{old}$ to $\mathbf{w}_{new}$ so repeatedly updating the parameters using

$$\mathbf{w} \leftarrow \mathbf{w} + \delta\frac{\partial\mathcal{L}}{\partial\mathbf{w}}$$

will yield an increase in the joint likelihood and so move $\mathbf{w}$ to the point of maximum posterior probability. The step size will govern the stability and speed of convergence to this point.

An alternative to steepest descent is to use the Newton method which you may have met in school maths as a technique for finding the roots of functions $f(x) = 0$ from an initial guess of x_0. The next guess is

$$x_{k+1} \leftarrow x_k - \frac{f(x_k)}{f'(x_k)}$$

Now we are looking for the stationary points $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{0}$ and so we can take the Newton method to find the roots of a single variable function and extend it to deal with multiple variables in which case the Newton routine we require is defined as

$$\mathbf{w} \leftarrow \mathbf{w} - \left(\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w} \partial \mathbf{w}^\mathsf{T}} \right)^{-1} \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$$

This will converge faster than the gradient based method but also may be prone to *overshooting* the maximum point.

Employing our expressions for the above terms then

$$\begin{aligned}
\mathbf{w} &\leftarrow \mathbf{w} + \left(\mathbf{\Phi}^\mathsf{T} \mathbf{V} \mathbf{\Phi} + \frac{1}{\alpha} \mathbf{I} \right)^{-1} \left(\mathbf{\Phi}^\mathsf{T} \mathbf{t} - \mathbf{\Phi}^\mathsf{T} \mathbf{p} - \frac{1}{\alpha} \mathbf{w} \right) \\
&= \left(\mathbf{\Phi}^\mathsf{T} \mathbf{V} \mathbf{\Phi} + \frac{1}{\alpha} \mathbf{I} \right)^{-1} \left(\left(\mathbf{\Phi}^\mathsf{T} \mathbf{V} \mathbf{\Phi} + \frac{1}{\alpha} \mathbf{I} \right) \mathbf{w} + \mathbf{\Phi}^\mathsf{T} \mathbf{t} - \mathbf{\Phi}^\mathsf{T} \mathbf{p} - \frac{1}{\alpha} \mathbf{w} \right) \\
&= \left(\mathbf{\Phi}^\mathsf{T} \mathbf{V} \mathbf{\Phi} + \frac{1}{\alpha} \mathbf{I} \right)^{-1} \mathbf{\Phi}^\mathsf{T} \left(\mathbf{V} \mathbf{\Phi} \mathbf{w} + \mathbf{t} - \mathbf{p} \right)
\end{aligned}$$

and so at each step $\mathbf{w}$ is updated and using the new values of $\mathbf{w}$ then the elements of both $\mathbf{p}$ and $\mathbf{V}$ are updated after which the next Newton step is re-applied and this is continued until convergence.

2.4 Demonstration of Laplace Approximation

The Matlab code laplace_demo.m in the Week 4 Laboratory folder will enable you to reproduce the following diagrams. Consider two classes of object which are characterised by two features and (in the never to be repeated scenario in the real world) these features are distributed for each class such that

$$\begin{aligned} \mathbf{x}|C=1 &\sim \mathcal{N}([1\ ,\ 5], \mathbf{I}) \\ \mathbf{x}|C=0 &\sim \mathcal{N}([-5\ ,\ 1], 1.1\mathbf{I}) \end{aligned}$$

A random sample of 30 examples from each of the two classes is drawn from the above distributions and are plotted in Figure (2).

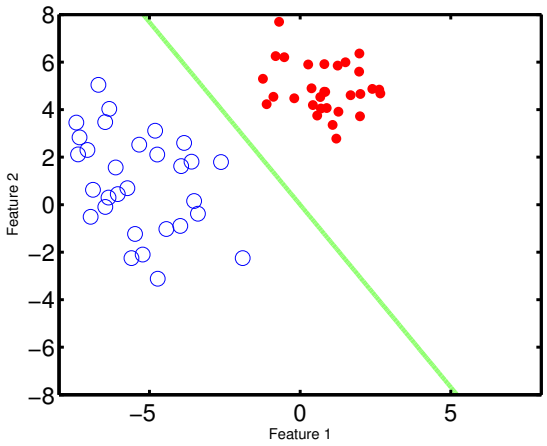

Figure 2: The blue circles are examples from class $C = 0$ and the solid red dots are examples from class $C = 1$. The green line shows the decision boundary $P(C = 1|\mathbf{x}) = 0.5$ obtained from the estimated $\mathbf{w}_{MAP}$ using the Newton routine described above.

These sixty data-points and the corresponding target values i.e. $t = 1$ if $\mathbf{x} \in C = 1$ and $t = 0$ if $\mathbf{x} \in C = 0$ are then used in the Newton method to identify $\mathbf{w}_{MAP}$. The prior was set a variance of $\alpha = 100$ which of course means that the prior will have a rather small effect on the likelihood in transforming it into the posterior. In other words there will be little in the

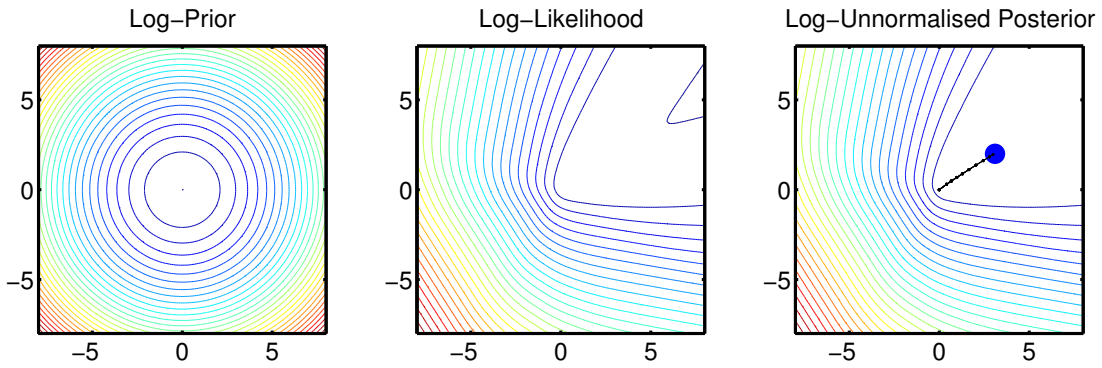

Figure 3: The three contour plots above show the negative logarithm of parameter probability distributions where the left-hand plot shows the distribution of the parameter values $\mathbf{w} = [w_1 \quad w_2]^\mathsf{T}$ under the defined prior. The middle plot shows the negative log-likelihood which is distinctly non-Gaussian and the right-hand plot shows the joint likelihood (un-normalised posterior). The large solid blue dot shows the point in parameter space where the posterior is a maximum and the lines of small dark dots shows the evolution of the Newton algorithm towards this point starting from and initial point of $\mathbf{w} = [0 \;\; 0]$, ten steps are required to achieve this optimum.

way of regularising effect of the prior. This is clear from the contour plots shown in Figure (3).

The same figure also shows how the Newton method finds the parameter values which yield the maximum of the posterior.

Of course being good Bayesians we are really interested in the approximation to the parameter posterior which this method provides us with. Why? because when we go on to make classification predictions we can average our uncertainty in the parameter estimates over this approximate posterior.

So the question is how good is the approximation? well in this particular case we can visualise the actual posterior alongside our Laplace approximation. Now remember that we are putting a multivariate-Gaussian onto the most probable *a posteriori* point in parameter space and then using the curvature of the posterior at this point to define the covariance of our Gaussian approximation.

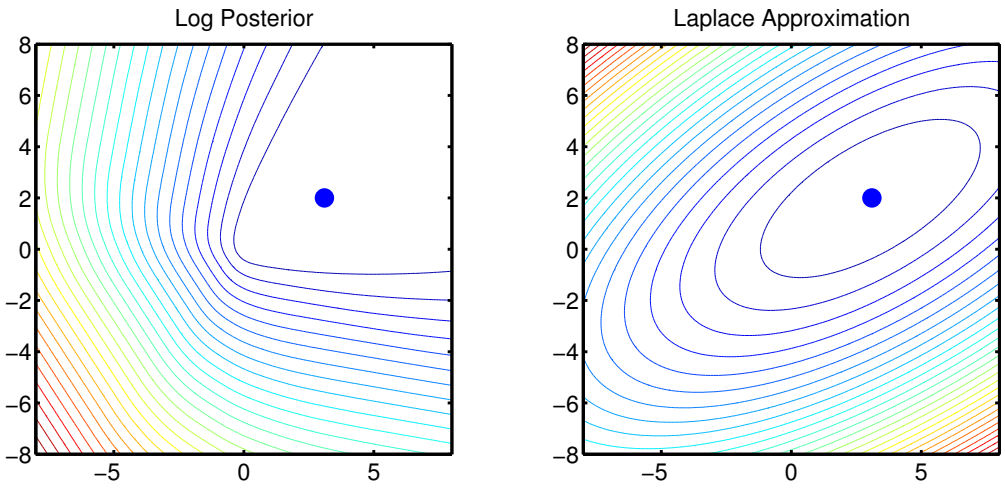

Figure 4: The left-plot shows the negative log-posterior whilst the right-plot shows the Laplace approximation. The first thing to note is that the location of the maximum has been reasonably well identified. The second point is to note that the positive curvature of the posterior (as both parameter values increase they become *a posteriori* more probable. We can observe this curvature in our Laplace approximation, however, note that as we move away from the *MAP* value the approximation is not so good.)

2.5 Logistic Regression Classification

So now by using the Newton method we can find the maximum of the posterior and with this the Laplace approximation to the posterior distribution can be employed.

Now to make predictions we want the following distribution

$$P(C = 1|\mathbf{x}_{new}, \alpha, \mathbf{X}, \mathbf{t}) = \int P(C = 1|\mathbf{x}_{new}, \mathbf{w}) p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha) d\mathbf{w}$$

We have approximated our posterior over the parameters $\mathbf{w}$, $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha)$ with a Gaussian via the Laplace approximation. So we can make a **Monte Carlo** estimate of the above integral using samples simulated from our approximate posterior such that

$$
\begin{aligned}
P(C=1|\mathbf{x}_{new},\alpha,\mathbf{X},\mathbf{t}) &\approx \frac{1}{N}\sum_{n_s=1}^{N_s} P(C=1|\mathbf{x}_{new},\mathbf{w}_s) \\
&= \frac{1}{N}\sum_{n_s=1}^{N_s} \frac{1}{1+\exp(-\mathbf{w}_s^{\mathsf{T}}\phi(\mathbf{x}_{new}))}
\end{aligned}
$$

where each $\mathbf{w}_s$ is simulated or drawn from the approximate Gaussian posterior, $\mathbf{w}_s \sim \mathcal{N}(\mathbf{w}_{MAP}, \mathbf{C})$ and this is easy to simulate (see laboratory sheet). of course we could go the whole road and use samples from the true posterior and to do this we have to resort to **Markov-Chain-Monte-Carlo** techniques, but for now we will just use our Laplace approximation to the parameter posterior.

The alternative to approximate Monte-Carlo averaging is to assume that the posterior is sharply peaked around the MAP value and so we can use the approximation simply uses the MAP estimate and so class predictions based on a new data point $\mathbf{x}_{new}$ are made by using the approximate predictive posterior probability given below

$$
\begin{aligned}
P(C=1|\mathbf{x}_{new},\alpha,\mathbf{X},\mathbf{t}) &\approx P(C=1|\mathbf{x}_{new},\mathbf{w}_{MAP},\alpha,\mathbf{X},\mathbf{t}) \\
&= \frac{1}{1+\exp(-\mathbf{w}_{MAP}^{\mathsf{T}}\phi(\mathbf{x}_{new}))}
\end{aligned}
$$

So the discriminant function we are using indicates that if $P(C=1|\mathbf{x}_{new},\alpha,\mathbf{X},\mathbf{t}) > 0.5$ then $\mathbf{x}_{new}$ is assigned to Class $C=1$ and $C=0$ otherwise.

2.6 Demonstration of Bayesian Logistic Regression Classification

The matlab file logistic_classification_demo.m will produce the following plots which show a two class problem based on two-dimensional feature vectors.

By using a simple polynomial basis such that

$$
\mathbf{\Phi} = \begin{bmatrix}
1 & x_{11} & x_{12} & x_{11}^2 & x_{12}^2 & \cdots & x_{11}^K & x_{12}^K \\
\vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
1 & x_{N1} & x_{N2} & x_{N1}^2 & x_{N1}^2 & \cdots & x_{N1}^K & x_{N1}^K
\end{bmatrix}
$$

where now $\mathbf{\Phi}$ is an $N \times (DK+1)$ dimensional matrix where in this case $D=2$. Note that we have only taken each feature to a polynomial power

and have not considered any cross-terms in the basis expansion such as for example $x_{11}x_{12}, x_{11}x_{12}^2, x_{11}^2x_{12}$.

Figure (5) shows the linear decision boundary learned using the methods just described (Laplace approximation of the model parameter posterior for a logistic regression model) for the two-dimensional data set. Figure (6) shows the more flexible decision boundary achieved when a $K = 3$ polynomial basis is used in the model.

We have normalised the coefficient values w_i with the diagonal terms of the covariance matrix $\mathbf{C}$ which will be the variance of the MAP weight values so clearly a small value indicates that it is not important in the model and its removal would amount to a negligble decrease in likelihood (perhaps an increase in predictive likelihood). Remember that the matrix of second-order partial derivative defines how the curvature of our likelihood varies at a particular point. So if the curvature in a particular direction corresponding to a specific parameter is small then perturbations to that parameter will have a very small effect on the actual function (likelihood) indicating that the parameter may not be relevant.

Do the normalised values of the weighting coefficients seem sensible?

Of course the number of classification errors made on an independent test test will be the real measure of what level of complexity is required for this model. In this weeks laboratory session we will explore the generalisation properties of this discriminative method of classification.

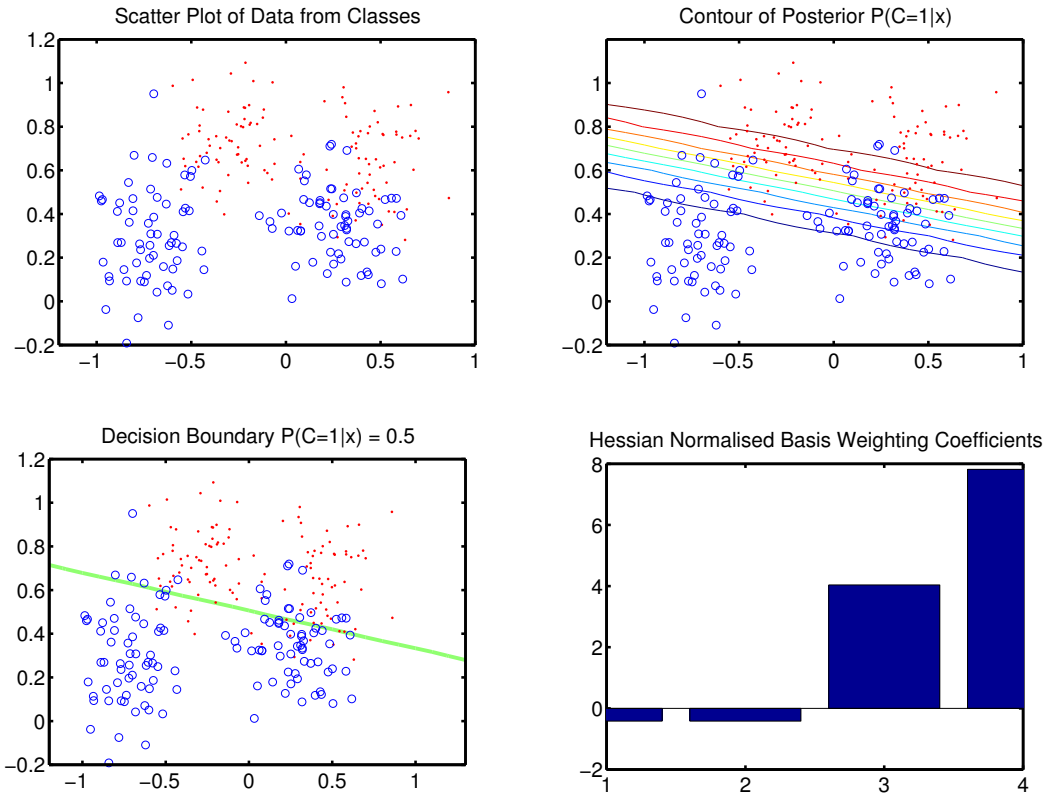

Figure 5: The top-left plot shows the two-dimensional data plotted as a scatter-plot with the two classes differentiated by dots and circles, note the classes overlap. The right hand plot shows the posterior probability of class membership when using a linear model i.e. $K = 1$) and the decision boundary $P(C = 1|\mathbf{x}) = 0.5$ is shown in the bottom left plot. The magnitude to the weighting coefficients normalised by the square-root of the Hessian matrix are shown in the bottom right plot, small values indicate that the weights might actually be zero and have little effect on the achieved data likelihood.

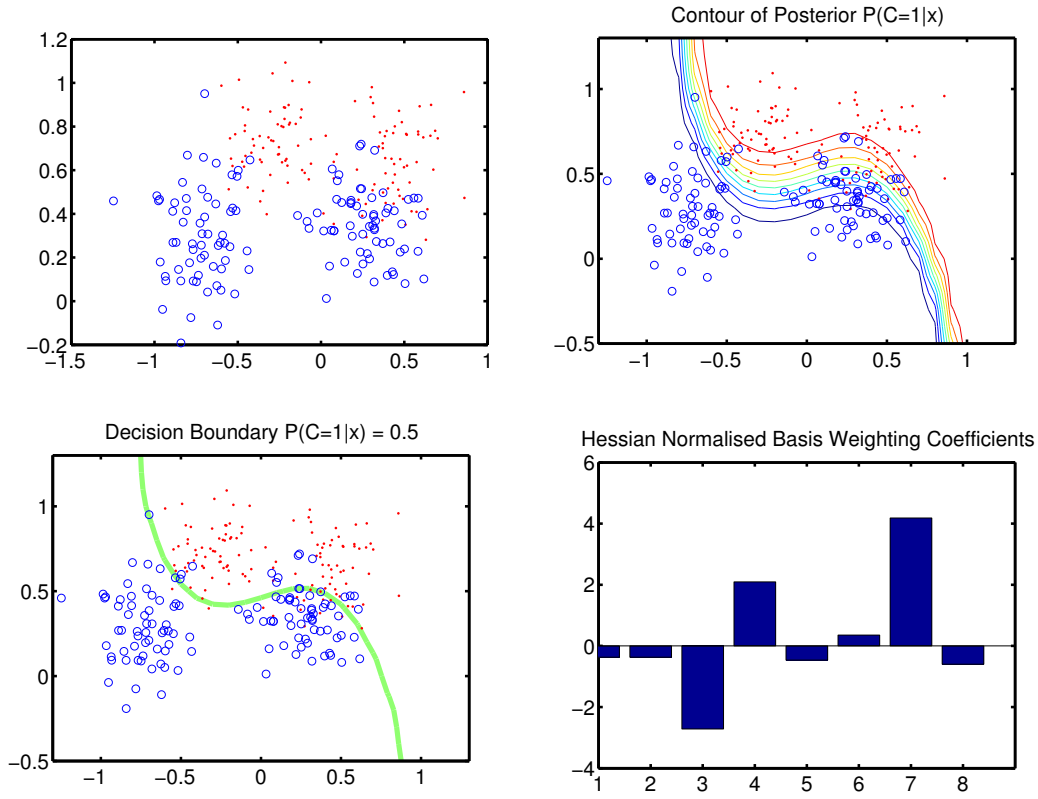

Figure 6: The top-left plot shows the two-dimensional data plotted as a scatter-plot with the two classes differentiated by dots and circles, note the classes overlap. The right hand plot shows the posterior probability of class membership when using a polynomial model of order $K = 3$ and the decision boundary $P(C = 1|\mathbf{x}) = 0.5$ is shown in the bottom left plot. The magnitude to the weighting coefficients normalised by the square-root of the Hessian matrix are shown in the bottom right plot, small values indicate that the weights might actually be zero and have little effect on the achieved data likelihood.

3 Generative Classification Methods

The previous approach to classification focused on modeling the discriminant function directly using a linear model i.e.

$$\log \frac{P(C=1|\mathbf{x})}{P(C=0|\mathbf{x})} = \mathbf{w}^\mathsf{T}\phi(\mathbf{x})$$

The generative approach on the other hand seeks to define the discriminant function by directly estimating the posterior ratio from the data likelihood and prior terms i.e.

$$\frac{P(C=1|\mathbf{x})}{P(C=0|\mathbf{x})} = \frac{P(\mathbf{x}|C=1)P(C=1)}{P(\mathbf{x}|C=0)P(C=0)}$$

Now given a training data set, $\mathbf{X}, \mathbf{t}$, we can estimate the prior probabilities of class membership by simply counting the numbers of instance of each class in the data and normalising by the total number of data samples i.e.

$$\widehat{P}(C=k) = \frac{1}{N_k}\sum_{n=1}^{N}\delta(t_n, k)$$

where $\delta(t_n, k)$ equals one if the target value t_n (the class label) corresponds to the kth class and N_k corresponds to the number of examples from class k. Note that the *hat* notation is being used to indicate that we are estimating the probability of class membership from this finite data sample.

Now we require the class conditional data-likelihood $P(\mathbf{x}|C=k)$, that is the probability density or distribution from which the data is generated.

3.1 Class Conditional Density & Distribution Estimates

Now the first thing to note here is that we need to make and estimates of the class conditional densities or distributions (if the features are discrete). We will look at this important and general problem, probability density estimation, in the first two lectures devoted to Unsupervised Learning. However, for now we will look at two specific situations where we can make assumptions about the parametric form of the class-conditional likelihoods.

3.1.1 Multivariate Gaussian Likelihood

Let us for now assume that we have reason to believe that our class-conditional likelihoods are well represented by multivariate Gaussians such that

$$p(\mathbf{x}|C=k) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}_k|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\mathsf{T}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

Then we require to obtain estimates for the mean vectors $\widehat{\boldsymbol{\mu}}_k$ and the covariance matrix $\widehat{\boldsymbol{\Sigma}}_k$ to obtain our estimated class-conditional likelihood $\widehat{p}(\mathbf{x}|C=k)$ which can be plugged into our discriminant function.

Lets expand the discriminant function for two classes, say k and l then it is easy to show that

$$\begin{aligned}\log\frac{P(C=k|\mathbf{x})}{P(C=l|\mathbf{x})} &= \log\frac{P(\mathbf{x}|C=k)}{P(\mathbf{x}|C=l)} + \log\frac{P(C=k)}{P(C=l)} \\ &= \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} + \mathbf{w}^\mathsf{T}\mathbf{x} + b_0\end{aligned}$$

where $\mathbf{A} = \boldsymbol{\Sigma}_l^{-1} - \boldsymbol{\Sigma}_k^{-1}$ and $\mathbf{w} = \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \boldsymbol{\Sigma}_l^{-1}\boldsymbol{\mu}_l$ with

$$b_0 = \log\frac{P(C=k)}{P(C=l)} + \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_l|}{|\boldsymbol{\Sigma}_k|} + \frac{1}{2}\left(\boldsymbol{\mu}_l^\mathsf{T}\boldsymbol{\Sigma}_l^{-1}\boldsymbol{\mu}_l - \boldsymbol{\mu}_k^\mathsf{T}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\right)$$

So what we can see is that the discriminant function that we obtain when assuming multivariate Gaussian class-conditional densities is a quadratic function of the features $\mathbf{x}$ and so we have a quadratic decision surface. It should also be clear that if a common covariance matrix across all classes is assumed then our discriminant reduces to a linear function of the form $\mathbf{w}^\mathsf{T}\mathbf{x} + b_0$ where $\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_l)$ which relies on the difference in the class means.

Now we have to estimate the parameters of the conditional-likelihood, in this case mean and covariances, to obtain the required posterior class probabilities. As we are really only interested in the discriminant function at the end of the day then it can be argued that most effort should focus on estimating a functional form for the posterior log-likelihood ratio as in the discriminative approach.

The generative approach on the other hand requires to make good estimates of the density to obtain the discriminant function and this can be

a weakness of the method in that requiring data from the regions of high density for each class to estimate parameter values e.g. mean values, may not necessarily help in defining the discriminant function. However despite this criticism generative methods for classification are useful in a number of situations.

3.2 Naive Bayes Classifier

In **Bioinformatics** microarray data can be used to build classifiers which will be capable of discriminating between cancerous and healthy tissue samples. Each sample is defined by the amount of mRNA that a large numbers of gene express in healthy or diseased conditions. Often there are over 30,000 genes, so this means that we have a feature vector $\mathbf{x} \in \mathbb{R}^D$ where $D = 30,000$. If we assume that the mRNA levels are roughly Gaussian then we can see that estimating $\mathbf{\Sigma}_{healthy}$ a $30,000 \times 30,000$ dimensional covariance matrix is going to be impossible given that the number of samples will be as small as several dozen.

So despite there possibly being features which will be correlated with each other it is impractical to even consider attempting to estimate a full covariance. So we are forced to make a further assumption that the covariance matrix is diagonal such that

$$\mathbf{\Sigma}_k = \begin{pmatrix} \sigma_1^2 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_3^2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & \sigma_{D-1}^2 & 0 \\ 0 & \cdots & \cdots & 0 & \sigma_D^2 \end{pmatrix}$$

In this case then the multivariate Gaussian reduces to a product form such that

$$p(\mathbf{x}|C = k) = \prod_{d=1}^{D} p(x_d|C_k) = \prod_{d=1}^{D} \mathcal{N}_{x_d}(\mu_d, \sigma_d)$$

Despite this form of classifier being referred to as **Naive Bayes** or **Idiots Bayes**, presumably because of the naive assumption of there being no covariance between features, in many applications such a classifier works surprisingly well.

One particular application within **Information Retrieval** is document classification which we shall look at briefly here.

3.3 Document Classification

Lets assume that we have a number of Documents d and they each have (or have not) the occurrence of words w from a dictionary $\mathcal{D}$. If we assume a simple bag-of-words document model then we can model the document as $|\mathcal{D}|$ single draws from a binomial distribution, such that for word w the probability of the word occurring in the document from class k is p_{kw} and the probability of it not occurring in the class k document is obviously $1 - p_{kw}$. It word w occurs in the document at least once then we assign the feature corresponding to the word the value 1 and if it does not occur in the document we assign the feature the value 0. So each document will be represented by a feature vector of ones and zeros with the same length as the size of the dictionary. Clearly for large dictionaries we will need to employ a Naive Bayes classifier. Let us say that we can create a matrix $\mathbf{D}$ whose rows correspond to each document and columns represent the dictionary terms so that the element $\mathbf{D}_{dw}$ indicates the presence or absence of the word w in document d. So using Naive Bayes then the class-conditional probability of a document d coming from class k is

$$p(\mathbf{D}_d|C = k) = \prod_{w=1}^{|\mathcal{D}|} p(\mathbf{D}_{dw}|C_k) = \prod_{w=1}^{D} p_{kw}^{\mathbf{D}_{dw}}(1 - p_{kw})^{1-\mathbf{D}_{dw}}$$

Once the parameter values p_{kw} are estimated then the estimate of the class conditional likelihood can be plugged into the discriminant function to make classification. We will see in subsequent lectures that the Maximum-Likelihood estimate for the parameters p_{kw} is simply

$$\widehat{p}_{kw} = \frac{1}{N_k} \sum_{d \in C_k} \mathbf{D}_{dw}$$

So if a term does not occur in the documents from class k then $\widehat{p}_{kw} = 0$ which seems a little pessimistic as it may be that additional documents from the class may well have the word. It is also somewhat inconvenient in that if $\widehat{p}_{kw} = 0$ for one word then $p(\mathbf{D}_d|C = k) = 0$ which makes no real sense.

In further lectures we will look at Bayesian estimates of distribution parameters and we will see for binary variables that the MAP estimator is a

more reasonable, and computationally convenient,

$$\widehat{p}_{kw} = \frac{1 + \sum_{d \in C_k} \mathbf{D}_{dw}}{2 + N_k}$$

4 Conclusion

We have looked at examples of discriminative and generative methods of classification. The discriminative method has its focus on the discriminative function whilst the generative approaches focus on making estimates of class-conditional distributions. For the discriminative method considered we have only looked at binary classification and extending logistic regression to multiple-classes requires a little more computational effort. on the other hand the generative methods are particularly simple and naturally deal with a multiplicity of classes. The laboratories this week will investigate both methods of classification.