3F8: Inference
Bayesian Non-Linearized Logistic Regression
Full Technical Report

Tommy Rochussen
Downing College
tnr22

April 1, 2022

Abstract

Probabilistic approaches to inference have the advantage that they use more information in preparing the inference and so can deliver more information to the user. Here we attempt to improve upon the logistic regression model (with non-linearized input features) of the regular coursework by describing each of the regression weights with a probability distribution, and learning the mean and variance for each weight after assuming independent Gaussian prior distributions for them with zero mean and unit variance. To get past issues of intractability, we use deterministic approximations (Laplace approximation, error function approximation to a logistic function) to great success. Bayesian model selection is briefly explored and we adpot a grid-search method to select the best of 100 models, with model evidence as the selection metric. The optimal model is explored and compared to other models.

Introduction

1. Given a dataset of two-dimensional input data and a binary output, the task is to develop a model that can accurately infer which class new input data belongs to. In the regular coursework, logistic regression is used but falls short as it is only capable of producing linear decision boundaries. It is improved by expanding the input features through Gaussian radial basis functions, thus non-linearizing the regression and producing non-linear decision boundaries.

2. We believe we can improve upon the inference of the non-linear logistic regression model of the regular coursework by adopting a probabilistic approach, namely through Bayesian Inference. This is likely to be a better approach as it makes use of all possible information including prior beliefs about logistic regression model weights before making the inference (the class predictions).

3. The issue is that there are a few intractable distributions and so we must approximate them. The posterior distribution over the model weights is intractable, and so the Laplace approximation is used to approximate the posterior distribution by a Gaussian distribution. Furthermore, calculating the predictive distribution is also intractable - we circumvent this by approximating a logistic function as a Gaussian CDF.

1 Theory and Derivations

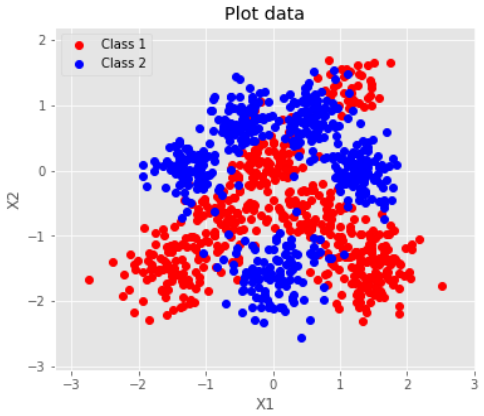

Figure 1: The binary dataset used

1.1 Laplace Approximation

The idea of the Laplace approximation is to fit a Gaussian distribution to the (intractable) true posterior distribution, since Gaussian posterior distributions are tractable. The Gaussian will be centered around the maximum a posteriori estimate for the parameters, that is to say centred around the parameters that maximise the posterior distribution, and it will have the same second derivative as the posterior distribution at this point too. These two factors determine the parameter means and covariance matrix of the Gaussian respectively.

Because we want to fit a Gaussian, is natural to assume a Gaussian prior distribution of the parameters, $p(w) = \mathcal{N}(\mathbf{w}|\mathbf{m_0}, \mathbf{S_0})$ where $\mathbf{m_0}$ and $\mathbf{S_0}$ are the prior mean vector and covariance matrix respectively. The log of the posterior distribution is then $ln(p(\mathbf{w}|\mathbf{t})) = -\frac{1}{2}(\mathbf{w}-\mathbf{m_0})^T\mathbf{S_0}^{-1}(\mathbf{w}-\mathbf{m_0})+\sum_{n=1}^{N}(t_n ln(y_n)+(1-t_n)ln(1-y_n))$, where $y_n = \sigma(\mathbf{w}^T\boldsymbol{\phi}_n)$ and $\mathbf{t}$ is the true class label vector. Obtaining the Gaussian approximation, we maximise the posterior distribution to get w_{MAP}, which will define the mean of the Gaussian approximation. This optimisation is done using the L-BFGS-B algorithm, which, being a gradient-based optimisation algorithm, uses the gradient of the posterior distribution. The inverse of the covariance matrix is found by the Hessian of the negative log likelihood, which is given by $\mathbf{S_N}^{-1} = -\nabla\nabla \ln p(\mathbf{w}|\mathbf{t}) = \mathbf{S_0}^{-1}+\sum_{n=1}^{N} y_n(1-y_n)\boldsymbol{\phi}_n\boldsymbol{\phi}_n^T$. This makes intuitive sense as the second derivative at an extremum of a function is a measure of how sharp or rounded the extremum is, meaning if the extremum is more rounded (smaller second derivative, middle term) then this corresponds to a high variance (inverse of left term) as there is a wider range from which the best value might be chosen. Note these mathematical results are from *Pattern Recognition and Machine Learning, Christoper Bishop*, p213-220.

Thus the log of the posterior distribution, the gradient of the log of the posterior, and the hessian of the log of the posterior need to be computed. All of these have a contribution from the Gaussian prior and a contribution from the likelihood. The log of the posterior distribution has constant terms in addition, but these can be ignored as they do not affect the MAP estimate of parameters, or indeed the hyper-parameters during the model evidence optimisation later on. To calculate the log gradient of the posterior, we consider first the log-likelihood term, followed by the log of the prior term. The gradient of the log-likelihood is derived

as follows:

$$\frac{\partial \mathcal{L}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \frac{\partial}{\partial \boldsymbol{\beta}}(log(P(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta})))$$
$$= \frac{\partial}{\partial \boldsymbol{\beta}}(\sum_{n=1}^{N} log(P(y^{(n)}|\tilde{\mathbf{X}}^{(n)})))$$
$$= \frac{\partial}{\partial \boldsymbol{\beta}}(\sum_{n=1}^{N} log(\sigma(\boldsymbol{\beta}^T\tilde{\mathbf{X}}^{(n)})^{y^{(n)}}) + \sum_{n=1}^{N} log((1-\sigma(\boldsymbol{\beta}^T\tilde{\mathbf{X}}^{(n)}))^{1-y^{(n)}})$$
$$= \sum_{n=1}^{N} \tilde{\mathbf{X}}^{(n)} \cdot y^{(n)}(1-\sigma(\boldsymbol{\beta}^T\tilde{\mathbf{X}}^{(n)})) + \sum_{n=1}^{N} \tilde{\mathbf{X}}^{(n)} \cdot (1-y^{(n)})(-)\sigma(\boldsymbol{\beta}^T\tilde{\mathbf{X}}^{(n)})$$
$$= \sum_{n=1}^{N} \tilde{\mathbf{X}}^{(n)}(y^{(n)} - \sigma(\boldsymbol{\beta}^T\tilde{\mathbf{X}}^{(n)}))$$

As a sanity check, if we minimise the difference between the true outputs, $y^{(n)}$, and the predictions, $\sigma(\boldsymbol{\beta}^T\tilde{\mathbf{X}}^{(n)})$, we are bringing the gradient of the log likelihood to zero. So we reach the maximum likelihood by making predictions that are the same as the true class labels, which is exactly what we need. Now considering the log of the prior term, the prior belief about the parameters is that they are Gaussian and have zero mean and σ_0 (hyper-parameter to be tuned later) variance, with no covariances - that is to say that $\mathbf{m_0} = \mathbf{0}$, and $\mathbf{S_0} = \sigma_0\mathbf{I}$. It can be shown that the only non-constant term in the log of the Gaussian prior is given by $\mathcal{P}$. The derivation from there is shown below.

$$\mathcal{P} = -\frac{1}{2}(\mathbf{w}-\mathbf{w_0})^T{\mathbf{S_0}}^{-1}(\mathbf{w}-\mathbf{w_0})$$
$$= -\frac{1}{2}\mathbf{w}^T{\mathbf{S_0}}^{-1}\mathbf{w} + {\mathbf{m_0}}^T{\mathbf{S_0}}^{-1}\mathbf{w} - \frac{1}{2}{\mathbf{m_0}}^T{\mathbf{S_0}}^{-1}\mathbf{m_0}$$
$$\therefore$$
$$\frac{d}{d\mathbf{w}}(\mathcal{P}) = -\frac{1}{2}({\mathbf{S_0}}^{-1}\mathbf{w} + ({\mathbf{S_0}}^{-1})^T\mathbf{w}) + ({\mathbf{S_0}}^{-1})^T\mathbf{m_0} - 0$$

but ${\mathbf{S_0}}^{-1}$ is symmetric as it is just a scaled identity matrix $\frac{1}{\sigma_0}\mathbf{I}$, so

$$\frac{d}{d\mathbf{w}}(\mathcal{P}) = -{\mathbf{S_0}}^{-1}\mathbf{w} + {\mathbf{S_0}}^{-1}\mathbf{m_0}$$

and in our case $\mathbf{m_0} = \mathbf{0}$, so

$$\frac{d}{d\mathbf{w}}(\mathcal{P}) = -{\mathbf{S_0}}^{-1}\mathbf{w}$$

Thus the gradient of the log of the posterior is given by $\frac{d}{d\mathbf{w}}(\ln p(\mathbf{w}|\mathbf{t})) = \sum_{n=1}^{N} \tilde{\mathbf{X}}^{(n)}(y^{(n)} - \sigma(\boldsymbol{\beta}^T\tilde{\mathbf{X}}^{(n)})) - {\mathbf{S_0}}^{-1}\mathbf{w} + {\mathbf{S_0}}^{-1}\mathbf{m_0}$. This is used in the optimising function to find $\mathbf{w_{MAP}}$, and so then the Gaussian approximation is given by $q(w) = \mathcal{N}(\mathbf{w}|\mathbf{w_{MAP}}, \mathbf{S_N})$.

1.2 Predictive Distribution

The predictive distribution for a new input feature $\boldsymbol{\phi}$ is given by

$$p(\mathcal{C}_1|\boldsymbol{\phi}, \mathbf{t}) = \int \sigma(a)\mathcal{N}(a|\mu_a, \sigma_a^2)da$$

where $a = \mathbf{w}^T\boldsymbol{\phi}$, $\mu_a = \mathbf{w}_{MAP}^T\boldsymbol{\phi}$, and $\sigma_a^2 = \boldsymbol{\phi}^T\mathbf{S_N}\boldsymbol{\phi}$. This is intractable, but it can be solved by another approximation. The expression represents the convolution of a Gaussian with a logistic sigmoid function, and if the logistic function is approximated by an error function (another sigmoidal function, the Gaussian CDF), the convolution becomes tractable. In Bishop's *Pattern Recognition and Machine Learning*, they speak of this as a probit function, but in most modern contexts the probit function refers to the inverse Gaussian CDF. To be clear, the function we use to approximate the logistic function is given by:

$$\Phi(x) = \frac{1}{2\sqrt{\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}dt = \frac{1}{2}erf(\sqrt{2}x) + \frac{1}{2}$$

We approximate the logistic function by selecting a horizontal scaling parameter λ. To find the scaling factor we force the gradients to be equal at the center as follows:

$$\frac{d}{da}(\frac{1}{2}erf(\lambda a) + \frac{1}{2})|_{a=0} = \frac{d}{da}(\frac{1}{1+e^{-a}})|_{a=0}$$
$$\lambda \cdot \frac{1}{2} \cdot \frac{2}{\sqrt{\pi}} = \frac{1}{4}$$
$$\lambda = \frac{\sqrt{\pi}}{4}$$

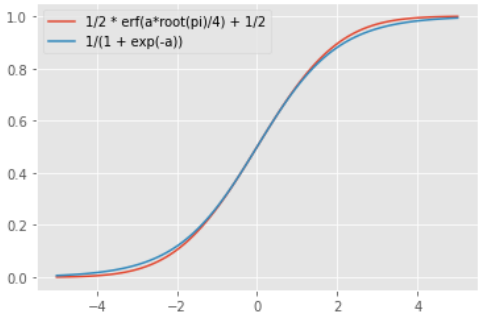

Figure 2: Error function approximation to a logistic function

Note this is different to the λ found in Bishop's *Pattern Recognition and Machine Learning* as their "probit" function is the standard Gaussian CDF (see p211), which as seen above is equivalent to a vertically scaled and shifted error function with a horizontal scaling factor of $\sqrt{2}$ - this explains why our λ is a factor of $\sqrt{2}$ larger than theirs. The convolution of this approximation and a Gaussian is a another horizontally scaled version of the approximation function, and that approximation function is in turn a close approximation of another horizontally scaled logistic function, whose scaling can be calculated by a similar gradient-matching method. This gives a predictive distribution of:

$$p(\mathcal{C}_1|\boldsymbol{\phi}, \mathbf{t}) \approx \sigma(\frac{\mu_a}{\sqrt{1 + \frac{\pi\sigma_a^2}{8}}})$$

1.3 Model Evidence

The approximation for the log of the model evidence is given by

$$\ln p(\mathcal{D}) \approx \ln p(\mathcal{D}|\mathbf{w_{MAP}}) + \ln p(\mathbf{w_{MAP}}) + \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{S_n}^{-1}|$$

The first term is the likelihood of the weights, the second term is the probability of the weights which comes from the prior (collectively these two terms come from the posterior), the third term is a constant, while the

fourth term represents the negative log of the determinant of the inverse covariance matrix, which increases if the determinant of the covariance matrix increases. Later we use this as an objective function to maximise in order to find the best model. There is some subtlety to this and why we use this instead of just maximising the log of the posterior distribution. Of course, maximising the posterior gives us a better fit of weights for the dataset - this is well understood, and this explains the first two terms. The interesting part is in the fourth term. The determinant of the covariance serves as a generalized variance, a scalar representation of the variance matrix. Maximising this serves to broaden the distribution, which is a way of stopping overfitting and keeping the model robust to new inputs. Maximising the evidence then serves to find the model with the best trade-off between being overtrained and undertrained.

2 Maximum A Posteriori vs Full Distribution

Here we construct two models in the exact same way, except in one of them the weights are set as the MAP weights, and in the other we maintain the probability distribution over the weights. They make identical use of the prior belief distribution, and both have their inputs put through radius 0.1 Gaussian radial basis functions.

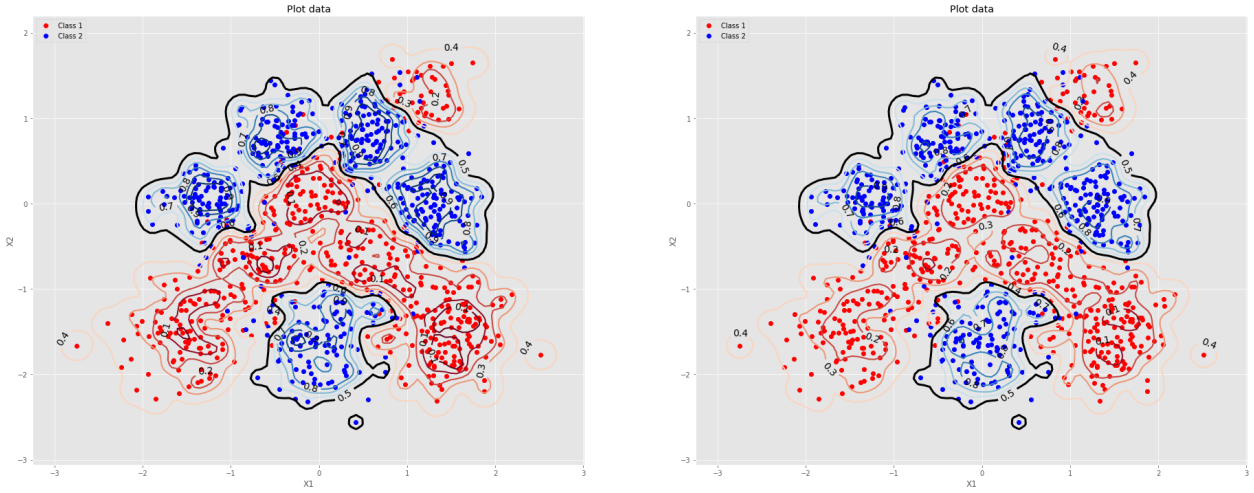

Figure 3: Plots showing data and contour lines for the predictive distribution generated by the Laplace approximation (left) and the MAP solution (right). Hyper-parameters set to $l = 0.1$ for both models, $\sigma_0 = 1.0$ for Bayesian model.

We see very similar predictive distributions for the full Bayesian and MAP approaches. The black contour is the decision boundary for class predictions, which seems at first glance to be identical for each approach. Visually, the difference is in how tight the contours are to each other - we see that the contours for the Bayesian predictions are slightly further apart implying less overfitting to the data and more gradual changes in the probability field, which is desirable as it means the Bayesian model is "less sure" of predictions which are near the decision boundary than the MAP point-estimated parameters model.

Avg. Train ll	**Avg. Test ll**
-0.2303	-0.2604

Table 1: Log-likelihoods for MAP solution.

Avg. Train ll	**Avg. Test ll**
-0.2848	-0.3059

Table 2: Log-likelihoods for Laplace approximation.

As usual we see test log-likelihoods that are slightly larger than their training dataset counterparts but perhaps the test log-likelihoods are slightly larger than they should be, possibly indicating some degree of

overfitting which may be caused by the hyper-parameters chosen, but if this is the case it is only very slight as the discrepancy is not necessarily extraordinary. The log-likelihoods for the full Bayesian approach are slightly smaller than those of the MAP approach, but this is to be expected as the Bayesian approach is stopped from overfitting by the prior, and so will naturally have less optimised log-likelihoods.

		$\hat{y}$	
		0	1
y	0	0.47	0.025
	1	0.05	0.455

Table 3: Conf. matrix for for MAP solution.

		$\hat{y}$	
		0	1
y	0	0.47	0.025
	1	0.05	0.455

Table 4: Conf. matrix for Laplace approximation.

We see the confusion matrices are identical - this assists in confirming that the decision boundaries (black contours) are indeed identical. This is to be expected since the mean of the weights in the Bayesian approach is chosen to be the MAP estimate for the weights - the means of the weights in the Bayesian model are the weights used in the MAP model, so the 0.5 probability contour should be identical. The difference in the approaches is that the Bayesian approach delivers a probability distribution around those weights, which softens the slope of the probability field around the decision boundary, but does not move the boundary itself in any way.

3 Selecting Optimal Model Hyper-Parameters

To compare many Bayesian models and select the best one, we choose the one with the greatest log model evidence for reasons discussed in Section 1.3. To come up with the different models, we adopt a 10x10 grid search approach where 10 values of Gaussian radial basis function radii (/variances) are each paired with 10 values representing the prior belief of the weight probability distribution standard deviation. The model evidence is calculated for each of the 100 resulting models, and the model corresponding to the largest evidence is selected as the best model.

It feels natural to space the values for both hyper-parameters logarithmically, as was done in the regular coursework for l. The grid search is initially run over a wide range of l and σ in order to locate the maximum in the objective function surface, and then a couple times over smaller ranges to hone in on the optimum more precisely. The optimal hyper-parameters found were $\sigma_{opt} = 1.015$, $l_{opt} = 0.37$, and the full grid of model evidences is shown as a heatmap in Figure 4. σ_{opt} is very similar to the one used in the model above, but l_{opt} is larger. As such we expect to see the optimal model to have smoother contours than those of the model above.

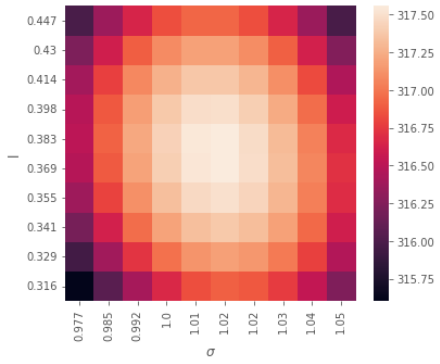

Figure 4: Heat map plot of the the approximation of the model evidence obtained in the grid search.

4 Optimal Bayesian Model

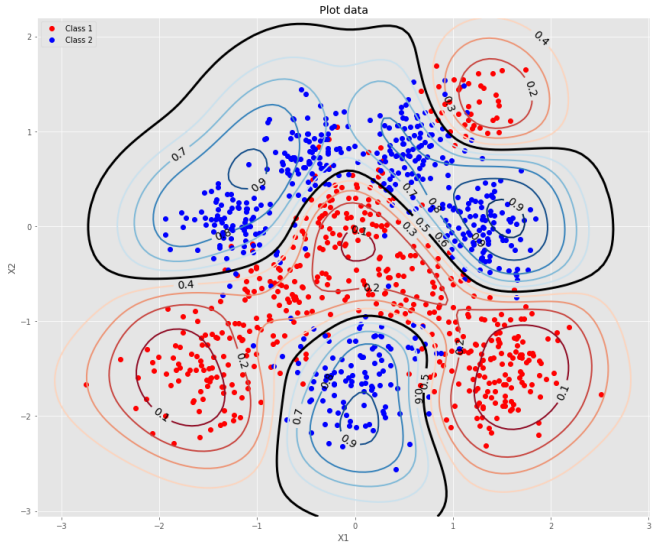

Figure 5: Visualisation of the contours of the class predictive probabilities for Laplace approximation after hyper-parameter tuning by maximising the model evidence.

We see that the probability contours of the optimal Bayesian model are much further apart than those of the model in Section 2, as well as smoother contours as predicted above. As discussed in Section 1.3, selecting models by optimising the model evidence is equivalent to finding the optimal degree of fit to the training data, which is why the contours of this model are smoother and further apart - a model that is overfitted to the data would twist and turn wildly to have all test datapoints on the correct sides of the decision boundary, and it would be very confident in the predictions made. A larger σ_0 corresponds to contours that are further apart, while a larger l corresponds to contours that have smoother shapes. This indicates that in the model with $l = 0.1$, $\sigma_0 = 1.0$ there was a slight degree of overfitting, since the probability contours were both closer and more meandering than in this model. This model only assigns predictions of high probability to datapoints that are well away from the decision boundary, and keeps prediction probabilities close to 0.5 for datapoints near the decision boundary to a greater degree than the model in Section 2.

Avg. Train ll	**Avg. Test ll**
-0.279	-0.305

Table 5: Average training and test log-likelihoods for Laplace approximation after hyper-parameter tuning by maximising the model evidence.

		$\hat{y}$	
		0	1
y	0	0.475	0.03
	1	0.05	0.445

Table 6: Confusion matrix for Laplace approximation after hyper-parameter tuning by maximising the model evidence.

We see the log-likelihoods for train and test datasets are closer together than in the model in Section 2, also implying a lesser degree of overfitting in this model. The confusion matrix is highly similar, even slightly worse here than in the earlier models. This highlights the advantage of Bayesian inference as a whole; it does not necessarily produce models that achieve superior categorical accuracy in classification than other methods, but it does produce models that are not confident about predicting incorrect classes rather than models that confidently predict incorrectly, and this is a great advantage.

Conclusions

1. Bayesian logistic regression has the advantage of extra information in the form of the prior belief about weights, and this advantage expresses itself in that the model holds back on overfitting and becoming overconfident near the decision boundary, even if the decision boundary itself is the same as that of the MAP weights model

2. Using the model evidence as an objective function for finding the optimal Bayesian model optimises for the tradeoff between maximising the posterior distribution (fitting the parameters to the data) and maximising the generalised variance of the posterior distribution (holding back on overfitting), which is equivalent to finding the model that is most in the sweet-spot between underfitting and overfitting

3. Despite a number of intractable expressions and having to deploy the Laplace approximation to the posterior distribution, the optimal Bayesian logistic regression model achieved a categorical accuracy of 92.0%

4. Despite such high performance, accuracy is not the main advantage of Bayesian inference, it is the more useful lack of certainty around the decision boundary which sets Bayesian inference apart from other methods. Applications which only require class predictions without probabilities are not suited to Bayesian inference, since enforcing a binary decision boundary undoes any of the work put into smoothing the probability field around the decision boundary, and the exact same binary decision boundary can be achieved through the model with MAP weights and no distribution around them