Bayesian logistic regression

So far we have only performed probabilistic inference in two particularly tractable situations: 1) small discrete models: inferring the class in a Bayes classifier, the card game, the robust logistic regression model. 2) "linear-Gaussian models", where the observations are linear combinations of variables with Gaussian beliefs, to which we add Gaussian noise.

For most models, we cannot compute the equations for making Bayesian predictions exactly. Logistic regression will be our working example. We'll look at how Bayesian predictions differ from regularized maximum likelihood. Then we'll look at different ways to approximately compute the integrals.

1 Logistic regression

As a quick review, the logistic regression model gives the probability of a binary label given a feature vector:

$$P(y\!=\!1\,|\,\mathbf{x},\mathbf{w}) \;=\; \sigma(\mathbf{w}^\top\mathbf{x}) \;=\; 1/(1+e^{-\mathbf{w}^\top\mathbf{x}}). \tag{1}$$

We usually add a bias parameter b to the model, making the probability $\sigma(\mathbf{w}^\top\mathbf{x}\!+\!b)$. Although the bias is often dropped from the presentation, to reduce clutter. We can always work out how to add a bias back in, by including a constant element in the input features $\mathbf{x}$.

You'll see various notations used for the training data $\mathcal{D}$. The model gives the probability of a vector of outcomes $\mathbf{y}$ associated with a matrix of inputs X (where the nth row is $\mathbf{x}^{(n)\top}$). Maximum likelihood fitting maximizes the probability:

$$P(\mathbf{y}\,|\,X,\mathbf{w}) = \prod_n \sigma(z^{(n)}\mathbf{w}^\top\mathbf{x}^{(n)}), \qquad \text{where } z^{(n)} = 2y^{(n)}\!-\!1, \;\text{ if }\; y^{(n)} \in \{0,1\}. \tag{2}$$

For compactness, we'll write this likelihood as $P(\mathcal{D}\,|\,\mathbf{w})$, even though really only the outputs $\mathbf{y}$ in the data are modelled. The inputs X are assumed fixed and known.

Logistic regression is most frequently fitted by a regularized form of maximum likelihood. For example L2 regularization fits an estimate

$$\mathbf{w}^* = \mathop{\arg\max}_{\mathbf{w}} \left[\log P(\mathbf{y}\,|\,X,\mathbf{w}) - \lambda\mathbf{w}^\top\mathbf{w}\right]. \tag{3}$$

We find a setting of the weights that make the training data appear probable, but discourage fitting extreme settings of the weights, that don't seem reasonable. Usually the bias weight will be omitted from the regularization term.

Just as with simple linear regression, we can instead follow a Bayesian approach. The weights are unknown, so predictions are made considering all possible settings, weighted by how plausible they are given the training data.

2 Bayesian logistic regression

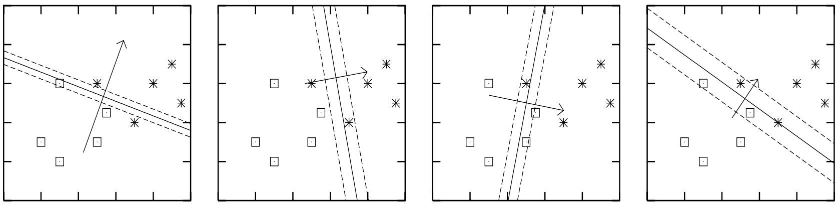
The posterior distribution over the weights is given by Bayes' rule:

$$p(\mathbf{w}\,|\,\mathcal{D}) = \frac{P(\mathcal{D}\,|\,\mathbf{w})\,p(\mathbf{w})}{P(\mathcal{D})} \propto P(\mathcal{D}\,|\,\mathbf{w})\,p(\mathbf{w}). \tag{4}$$

The normalizing constant is the integral required to make the posterior distribution integrate to one:

$$P(\mathcal{D}) = \int P(\mathcal{D}\,|\,\mathbf{w})\,p(\mathbf{w})\;\mathrm{d}\mathbf{w}. \tag{5}$$

The figures below[1] are for five different plausible sets of parameters, sampled from the posterior $p(\mathbf{w} \,|\, \mathcal{D})$.[2] Each figure shows the decision boundary $\sigma(\mathbf{w}^\top \mathbf{x}) = 0.5$ for one parameter vector as a solid line, and two other contours given by $\mathbf{w}^\top \mathbf{x} = \pm 1$.



The axes in the figures above are the two input features x_1 and x_2. The model included a bias parameter, and the model parameters were sampled from the posterior distribution given data from the two classes as illustrated. The arrow, perpendicular to the decision boundary, illustrates the direction and magnitude of the weight vector.

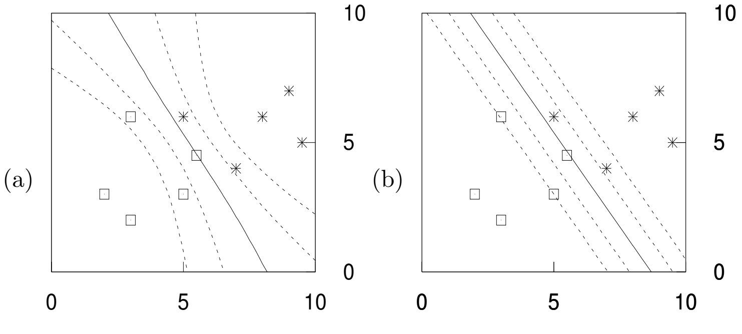
Assuming that the data are well-modelled by logistic regression, it's clear that we don't know what the correct parameters are. That is, we don't know what parameters we would fit after seeing substantially more data. The predictions given the different plausible weight vectors differ substantially.

The Bayesian way to proceed is to use probability theory to derive an expression for the prediction we want to make:

$$P(y \,|\, \mathbf{x}, \mathcal{D}) = \int p(y, \mathbf{w} \,|\, \mathbf{x}, \mathcal{D})\; \mathrm{d}\mathbf{w} \tag{6}$$

$$= \int P(y \,|\, \mathbf{x}, \mathbf{w})\, p(\mathbf{w} \,|\, \mathcal{D})\; \mathrm{d}\mathbf{w}. \tag{7}$$

That is, we should average the predictive distributions $P(y \,|\, \mathbf{x}, \mathbf{w})$ for different parameters, weighted by how plausible those parameters are, $p(\mathbf{w} \,|\, \mathcal{D})$. Contours of this predictive distribution, $P(y = 1 \,|\, \mathbf{x}, \mathcal{D}) \in \{0.5, 0.27, 0.73, 0.12, 0.88\}$, are illustrated in the left panel below. Predictions at some constant distance away from the decision boundary are less certain when further away from the training inputs. That's because the different predictors above disagreed in regions far from the data.



Again, the axes are the input features x_1 and x_2. The right hand figure shows $P(y = 1 \,|\, \mathbf{x}, \mathbf{w}^*)$ for some fitted weights $\mathbf{w}^*$. No matter how these fitted weights are chosen, the contours have to be linear. The parallel contours mean that the uncertainty of predictions falls at the same rate when moving away from the decision boundary, no matter how far we are from the training inputs.

[1] The two figures in this section are extracts from Figure 41.7 of MacKay's textbook (p499). Murphy's Figures 8.5 and 8.6 contain a similar illustration.

[2] It's not obvious how to generate samples from $p(\mathbf{w} \,|\, \mathcal{D})$, and in fact it's hard to do exactly. These samples were drawn approximately with a "Markov chain Monte Carlo" method.

It's common to describe L2 regularized logistic regression as MAP (Maximum a posteriori) estimation with a Gaussian $\mathcal{N}(0, \sigma_w^2 \mathbb{I})$ prior on the weights. The "most probable"[3] weights, coincide with an L2 regularized estimate:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \left[\log p(\mathbf{w} \,|\, \mathcal{D})\right] = \arg\max_{\mathbf{w}} \left[\log P(\mathcal{D} \,|\, \mathbf{w}) - \frac{1}{2\sigma_w^2}\mathbf{w}^\top \mathbf{w}\right]. \tag{8}$$

MAP estimation is *not* a "Bayesian" procedure. The rules of probability theory don't tell us to fix an unknown parameter vector to an estimate. We could view MAP as an approximation to the Bayesian procedure, but the figure above illustrates that it is a crude one: the Bayesian predictions (left) are qualitatively different to the MAP ones (right).

Unfortunately, we can't evaluate the integral for predictions $P(y \,|\, \mathbf{x}, \mathcal{D})$ in closed form. Making model choices for Bayesian logistic regression is also computationally challenging. The marginal probability of the data, $P(\mathcal{D})$, is the marginal likelihood of the model, which we might write as $P(\mathcal{D} \,|\, \mathcal{M})$ when we are evaluating some model choices $\mathcal{M}$ (such as basis functions and hyperparameters). We also can't evaluate the integral for $P(\mathcal{D})$ in closed form.

3 The logistic regression posterior is sometimes approximately Gaussian

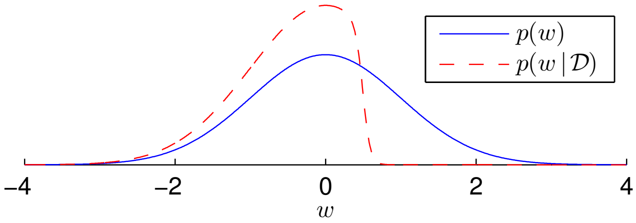
We're able to do some integrals involving Gaussian distributions. The posterior distribution over the weights $p(\mathbf{w} \,|\, \mathcal{D})$ is *not* Gaussian, but we can make progress if we can approximate it with a Gaussian.

Below is an example to illustrate how the posterior over the weights can look non-Gaussian. We have a Gaussian prior with one sigmoidal likelihood term. Here we assume we know the bias[4] is 10, and we have one datapoint with $y = 1$ at $x = -20$:

$$p(w) \propto \mathcal{N}(w; 0, 1) \tag{9}$$

$$p(w \,|\, \mathcal{D}) \propto \mathcal{N}(w; 0, 1)\, \sigma(10 - 20w). \tag{10}$$

We are now fairly sure that the weight isn't a large positive value, because otherwise we'd have probably seen $y = 0$. We (softly) slice off the positive region[5] and renormalize to get the posterior distribution illustrated below:



The distribution is asymmetric and so clearly not Gaussian. Every time we multiply the posterior by a sigmoidal likelihood, we softly carve away half of the weight space in some direction. While the posterior distribution has no neat analytical form, the distribution over plausible weights often does look Gaussian after many observations.

[The website version of this note has a question here.]

[3] "Most probable" is problematic for real-valued parameters. Really we are picking the weights with the highest probability density. But those weights aren't well-defined, because if we consider a non-linear reparameterization of the weights, the maximum of the pdf will be in a different place. That's why we prefer to describe estimating the weights as "regularized maximum likelihood" or "penalized maximum likelihood" rather than MAP.

[4] Perhaps we have many datapoints and have fitted the bias precisely, but we have one datapoint that has a novel feature turned on, and the example is showing the posterior over the weight that interacts with that one feature.

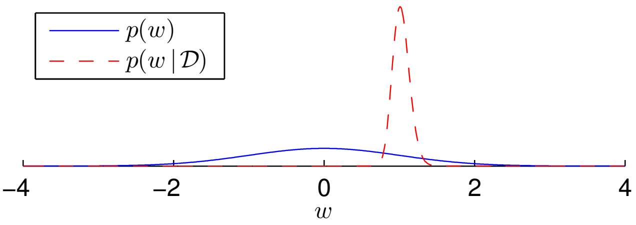
[5] If it's not obvious what's going on, plot $\sigma(10 - 20w)$ against w. We are multiplying our prior by this soft step function, which multiplies the prior by nearly one on the left, and nearly zero on the right.

As another example, let's consider $N = 500$ labels, $\{z^{(n)}\}$, generated from a logistic regression model with no bias and with $w = 1$ at $x^{(n)} \sim \mathcal{N}(0, 10^2)$. Then,

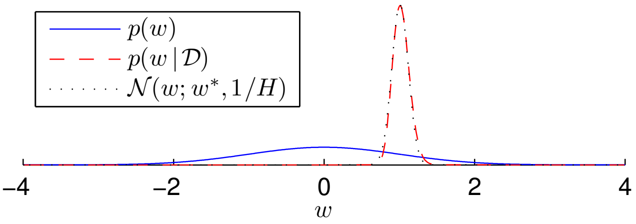
$$p(w) \propto \mathcal{N}(w; 0, 1) \tag{11}$$

$$p(w \,|\, \mathcal{D}) \propto \mathcal{N}(w; 0, 1) \prod_{n=1}^{500} \sigma(wx^{(n)}z^{(n)}), \qquad z^{(n)} \in \{\pm 1\}. \tag{12}$$

The posterior now appears to be a beautiful bell-shape:



Fitting a Gaussian distribution (using the Laplace approximation, next note) shows that the distribution isn't quite Gaussian... but it's close:


$p(w)$
$p(w \,|\, \mathcal{D})$
$\mathcal{N}(w; w^*, 1/H)$
−4
−2
0
2
4
w