Until now we have always worked with likelihoods and prior distributions that were conjugate to each other, allowing the computation of the posterior distribution to be done in closed form. Unfortunately, there are numerous situations where this will not be the case, forcing us to approximate the posterior and related quantities (such as the model evidence or expectations under the posterior distribution). Logistic regression is a common linear method for binary classification, and attempting to use the Bayesian approach directly will be intractable.

Logistic Regression

In linear regression, we supposed that were interested in the values of a real-valued function $y(\mathbf{x})\colon \mathbb{R}^d \to \mathbb{R}$, where $\mathbf{x}$ is a d-dimensional vector-valued input. Here, we will consider a similar setup, but with a twist: we restrict the output of the function y to the discrete space $y \in \{0, 1\}$. In machine learning, problems of this form fall under the category of **binary classification:** given an input $\mathbf{x}$, we wish to **classify** it into one of two categories, in this case denoted arbitrarily by 0 and 1.

We again assume that we have made some observations of this mapping, $\mathcal{D} = \bigl\{(\mathbf{x}_i, y_i)\bigr\}_{i=1}^{n}$, to serve as training data. Given these examples, the goal of binary classification is to be able to predict the label at a new input location $\mathbf{x}^*$.

As in linear regression, the problem is not yet well-posed without some restrictions on y. In linear regression, we assumed that the relationship between $\mathbf{x}$ and y was "mostly" linear:

$$y(\mathbf{x}) = \mathbf{x}^\top \mathbf{w} + \varepsilon(\mathbf{x}),$$

where $\mathbf{w} \in \mathbb{R}^d$ is a vector of parameters, and $\varepsilon(\mathbf{x})$ is the residual. This assumption is not very desirable in the classification case, where the outputs are restricted to $\{0, 1\}$ (note, for example, that $\mathbf{x}^\top \mathbf{w}$ is unbounded as the norm of $\mathbf{x}$ increases, forcing the residuals to grow ever larger).

In linear classification methods, we instead assume that the class-conditional probability of belonging to the "1" class is given by a nonlinear transformation of an underlying linear function of $\mathbf{x}$:

$$\Pr(y = 1 \mid \mathbf{x}, \mathbf{w}) = \sigma(\mathbf{x}^\top \mathbf{w}),$$

where σ is a so-called "sigmoid" (or "s-shaped") increasing function mapping the real line to valid probabilities in $(0, 1)$. The most-commonly used functions σ are the **logistic function:**

$$\sigma(a) = \frac{\exp(a)}{1 + \exp(a)},$$

or the standard normal cumulative distribution function:

$$\sigma(a) = \Phi(a) = \int_{-\infty}^{a} \mathcal{N}(x; 0, 1^2)\,\mathrm{d}x.$$

These two choices are compared in Figure 1. The main qualitative difference is that the logistic function has slightly heavier tails than the normal CDF. Linear classification using the logistic function is called **logistic regression**; linear classification using the normal CDF is called **probit regression.** Logistic regression is more commonly encountered in practice. Notice that the linear assumption above combined with the logistic function sigmoid implies that the **log odds** are a linear function of the input $\mathbf{x}$:

$$\log \frac{\Pr(y = 1 \mid \mathbf{x}, \mathbf{w})}{\Pr(y = 0 \mid \mathbf{x}, \mathbf{w})} = \log \frac{\frac{\exp(\mathbf{x}^\top \mathbf{w})}{1+\exp(\mathbf{x}^\top \mathbf{w})}}{1 - \frac{\exp(\mathbf{x}^\top \mathbf{w})}{1+\exp(\mathbf{x}^\top \mathbf{w})}} = \log \frac{\exp(\mathbf{x}^\top \mathbf{w})}{1} = \mathbf{x}^\top \mathbf{w}.$$

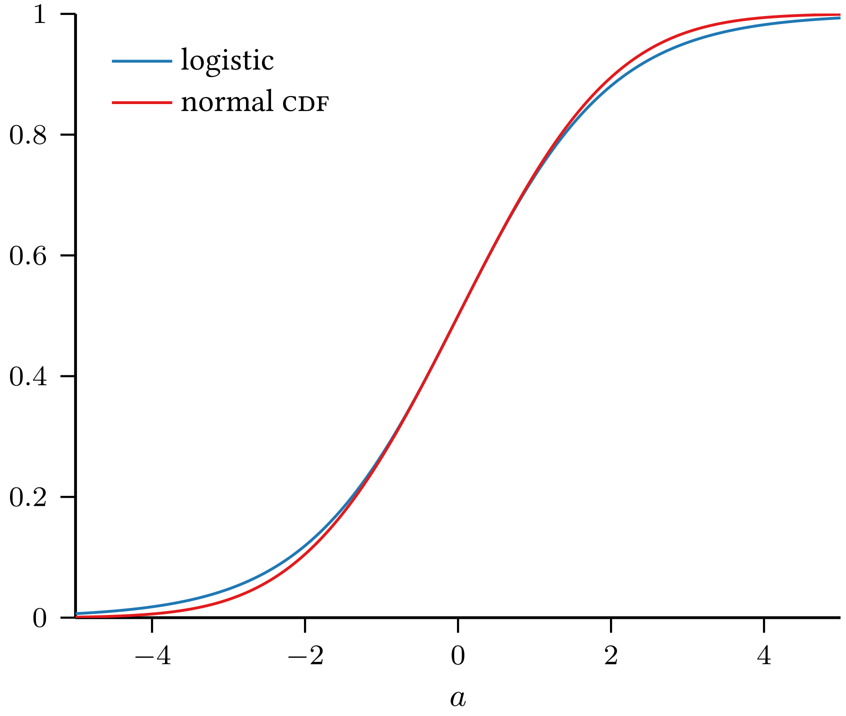

Figure 1: A comparison of the two sigmoid functions described in the text. The normal CDF curve in this example uses the transformation $\Phi(\sqrt{\frac{\pi}{8}}a)$, which ensures the slopes of the two curves are equal at the origin.

With the choice of the sigmoid function, and an assumption that our training labels $\mathbf{y}$ are generated independently given $\mathbf{w}$, we have defined our likelihood $\Pr(\mathbf{y} \mid \mathbf{X}, \mathbf{w})$:

$$\Pr(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \prod_{i=1}^{n} \sigma(\mathbf{x}_i^\top \mathbf{w})^{y_i} \bigl(1 - \sigma(\mathbf{x}_i^\top \mathbf{w})\bigr)^{1-y_i}. \tag{1}$$

To verify this equation, notice that each y_i will either be 0 or 1, so exactly one of y_i or $1 - y_i$ will be nonzero, which picks out the correct contribution to the likelihood.

The traditional approach to logistic regression is to maximize the likelihood of the training data as a function of the parameters $\mathbf{w}$:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \Pr(\mathbf{y} \mid \mathbf{X}, \mathbf{w});$$

$\hat{\mathbf{w}}$ is therefore a maximum-likelihood estimator (MLE). Unlike in linear regression, where there was a closed-form expression for the maximum-likelihood estimator, there is no such solution for logistic regression. Things aren't too bad though, because it turns out that for logistic regression the negative log-likelihood is convex and positive definite, which means there is a unique global minimum (and therefore a unique MLE). There are numerous off-the-shelf methods available for finding $\hat{\mathbf{w}}$: gradient descent, Newton's method, etc.

Bayesian logistic regression

A Bayesian approach to logistic regression requires us to select a prior distribution for the parameters $\mathbf{w}$ and derive the posterior distribution $p(\mathbf{w} \mid \mathcal{D})$. For the former, we will consider a multivariate Gaussian prior, identical to the one we used for linear regression:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

Now we apply Bayes' theorem to write down the desired posterior:

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{p(\mathbf{y} \mid \mathbf{X}, \mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y} \mid \mathbf{X}, \mathbf{w})p(\mathbf{w})\,\mathrm{d}\mathbf{w}} = \frac{p(\mathbf{y} \mid \mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y} \mid \mathbf{X})}.$$

Unfortunately, the product of the Gaussian prior on $\mathbf{w}$ and the likelihood (1), for either choice of sigmoid, does not result in a posterior distribution in a nice parametric family that we know. Likewise, the integral in the normalization constant (the model evidence) $p(\mathbf{y} \mid \mathbf{X})$ is intractable as well.

How can we proceed? There are two main approaches to continuing Bayesian inference in such a situation. The first is to use a deterministic method to find an approximation to the posterior (that will typically live inside some chosen parametric family). The second is to forgo a closed-form expression for the posterior and instead derive an algorithm to draw *samples* from the posterior distribution, which we may use to, for example, make Monte Carlo estimates to expectations. Here we will consider the **Laplace approximation,** an example of the first type of approach.

Laplace Approximation to the Posterior

Suppose we have an arbitrary parameter prior $p(\boldsymbol{\theta})$ and an arbitrary likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$, and wish to approximate the posterior

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{1}{Z} p(\mathcal{D} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}),$$

where the normalization constant Z is the unknown evidence. We define the following function:

$$\Psi(\boldsymbol{\theta}) = \log p(\mathcal{D} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}).$$

Ψ is therefore the logarithm of the *unnormalized* posterior distribution. The Laplace approximation is based on a Taylor expansion to Ψ around its maximum. First, we find the maximum of Ψ:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}).$$

Notice that the point $\hat{\boldsymbol{\theta}}$ is a **maximum a posteriori** (MAP) approximation to the parameters. Finding $\hat{\boldsymbol{\theta}}$ can be done in a variety of ways, but in practice it is usually fairly easy to find the gradient and Hessian of Ψ with respect to $\boldsymbol{\theta}$ and use some off-the-shelf optimization routine. This is an example of the saying "optimization is easier than integration."

Once we have found $\hat{\boldsymbol{\theta}}$, we make a second-order Taylor expansion to Ψ around this point:

$$\Psi(\boldsymbol{\theta}) \approx \Psi(\hat{\boldsymbol{\theta}}) - \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \mathbf{H}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}),$$

where $\mathbf{H}$ is the Hessian of the negative log posterior evaluated at $\hat{\boldsymbol{\theta}}$:

$$\mathbf{H} = -\nabla\nabla\Psi(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}.$$

Notice that the first-order term in the Taylor expansion vanishes because we expand arounded a maximum, where the gradient is zero. Exponentiating, we may derive an approximation to the (unnormalized) posterior distribution:

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \overset{\propto}{\sim} \exp\bigl(\Psi(\hat{\boldsymbol{\theta}})\bigr) \exp\biggl(-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \mathbf{H}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\biggr), \tag{2}$$

which we recognize as being proportional to a Gaussian distribution! The Laplace approximation therefore results in a normal approximation to the posterior distribution:

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \approx \mathcal{N}(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}, \mathbf{H}^{-1}).$$

The approximation is a Gaussian centered on the mode of the posterior, $\hat{\boldsymbol{\theta}}$, with covariance compelling the log of the approximation to match the curvature of the true log posterior at that point.

We note that the Laplace approximation also gives an approximation to the normalizing constant Z. In this case, it's simply a question of which normalizing constant we had to use to get (2) to normalize. A fairly straightforward calculation gives

$$Z = \int \exp\bigl(\Psi(\boldsymbol{\theta})\bigr)\,\mathrm{d}\boldsymbol{\theta} \approx \int \exp\bigl(\Psi(\hat{\boldsymbol{\theta}})\bigr) \exp\biggl(-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \mathbf{H}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\biggr)\,\mathrm{d}\boldsymbol{\theta} = \exp\bigl(\Psi(\hat{\boldsymbol{\theta}})\bigr)\sqrt{\frac{(2\pi)^d}{\det \mathbf{H}}},$$

where d is the dimension of $\boldsymbol{\theta}$.

Making Predictions

Suppose we have obtained a Gaussian approximation to the posterior distribution $p(\mathbf{w} \mid \mathcal{D})$; for example, applying the Laplace approximation above gives $p(\mathbf{w} \mid \mathcal{D}) \approx \mathcal{N}(\mathbf{w}; \hat{\mathbf{w}}, \mathbf{H}^{-1})$, where $\hat{\mathbf{w}}$ is the MAP approximation to the parameters and $\mathbf{H}$ is the Hessian of the negative log posterior evaluated at $\hat{\mathbf{w}}$.

Suppose now that we are given a test input $\mathbf{x}^*$ and wish to predict the binary label y^*. In the Bayesian approach, we marginalize the unknown parameters $\mathbf{w}$ to find the **predictive distribution**:

$$\Pr(y^* = 1 \mid \mathbf{x}^*, \mathcal{D}) = \int \Pr(y^* = 1 \mid \mathbf{x}^*, \mathbf{w}) p(\mathbf{w} \mid \mathcal{D})\,\mathrm{d}\mathbf{w} = \int \sigma(\mathbf{x}^{*T}\mathbf{w}) p(\mathbf{w} \mid \mathcal{D})\,\mathrm{d}\mathbf{w}.$$

Unfortunately, even with our Gaussian approximation to $p(\mathbf{w} \mid \mathcal{D})$, this integral cannot be evaluated if we use the logistic function in the role of the sigmoid σ. We can, however, compute the integral when using the normal CDF for σ:

$$\Pr(y^* = 1 \mid \mathbf{x}^*, \mathcal{D}) = \int \Phi(\mathbf{x}^{*T}\mathbf{w}) \mathcal{N}(\mathbf{w}; \hat{\mathbf{w}}, \mathbf{H}^{-1})\,\mathrm{d}\mathbf{w}.$$

This looks like an annoying d-dimensional integral, but notice that the vector $\mathbf{w}$ only appears in the expectation via the scalar product $\mathbf{x}^{*T}\mathbf{w}$. To proceed, we define the scalar value $a = \mathbf{x}^{*T}\mathbf{w}$ and rewrite this as

$$\Pr(y^* = 1 \mid \mathbf{x}^*, \mathcal{D}) = \int_{-\infty}^{\infty} \Phi(a) p(a \mid \mathcal{D})\,\mathrm{d}a.$$

Notice that a is a linear transformation of the Gaussian-distributed $\mathbf{w}$; therefore, a has a Gaussian distribution:

$$p(a \mid \mathcal{D}) = \mathcal{N}(a; \mu_{a|\mathcal{D}}, \sigma^2_{a|\mathcal{D}}),$$

where

$$\mu_{a\mid\mathcal{D}} = \mathbf{x}^{*T}\hat{\mathbf{w}}; \qquad \sigma^2_{a\mid\mathcal{D}} = \mathbf{x}^{*T}\mathbf{H}^{-1}\mathbf{x}^{*}.$$

Now we may finally compute the integral:

$$\Pr(y^* = 1 \mid \mathbf{x}^*, \mathcal{D}) = \int_{-\infty}^{\infty} \Phi(a)\, \mathcal{N}(a; \mu_{a\mid\mathcal{D}}, \sigma^2_{a\mid\mathcal{D}})\, \mathrm{d}a = \Phi\left(\frac{\mu_{a\mid\mathcal{D}}}{\sqrt{1 + \sigma^2_{a\mid\mathcal{D}}}}\right).$$

Notice that $\Phi(\mu_{a\mid\mathcal{D}})$ would be the estimate we would make using the MAP $\hat{\mathbf{w}}$ as a plug-in estimator. The $\sqrt{1 + \sigma^2_{a\mid\mathcal{D}}}$ term has the effect of making our prediction less confident (that is, closer to 1/2) according to our uncertainty in the value of $a = \mathbf{x}^{*T}\mathbf{w}$. This procedure is sometimes called **moderation,** because we force our predictions to be more moderate than we would have using a plug-in point estimate of $\mathbf{w}$.

We also note that if we only want to make point predictions of y^* using the 0–1 loss function, we only need to know which class is more probable (this was a general result from our discussion of Bayesian decision theory). In this case, the moderation has no effect on our ultimate predictions (you can check that we never change which side of 1/2 the final probability is), and we may instead simply find $\hat{\mathbf{w}}$. This is similar to the result we had in linear regression, where we could simply find the MAP estimator for $\mathbf{w}$ if we only ultimately cared about point predictions under a squared loss.

With loss functions different from the 0–1 loss, however, the uncertainty in $\mathbf{w}$ can indeed be important.