Computing logistic regression predictions using sampling based methods

In the previous note we approximated the logistic regression posterior with a Gaussian distribution. By comparing to the joint probability, we immediately obtained an approximation for the marginal likelihood $P(\mathcal{D})$ or $P(\mathcal{D} \mid \mathcal{M})$, which can be used to choose between alternative model settings $\mathcal{M}$, and we could use the Laplace approximation to make Bayesian prediction. We now look at other ways to make Bayesian predictions, not involving a Gaussian approximation.

1 Monte Carlo approximations

A route to avoiding Gaussian approximations is to approximate the prediction, which is an expectation, with an empirical average over samples:

$$P(y=1 \mid \mathbf{x}, \mathcal{D}) = \int \sigma(\mathbf{w}^\top \mathbf{x})\, p(\mathbf{w} \mid \mathcal{D})\, \mathrm{d}\mathbf{w} \tag{1}$$

$$= \mathbb{E}_{p(\mathbf{w} \mid \mathcal{D})}[\sigma(\mathbf{w}^\top \mathbf{x})] \tag{2}$$

$$\approx \frac{1}{S} \sum_{s=1}^{S} \sigma(\mathbf{w}^{(s)\top} \mathbf{x}), \quad \mathbf{w}^{(s)} \sim p(\mathbf{w} \mid \mathcal{D}). \tag{3}$$

Our prediction is the average of the predictions made by S different plausible model fits, sampled from the posterior distribution over parameters.

However, it is not at all obvious how to draw samples from the posterior over weights for general models. For simple versions of linear regression, we know that $p(\mathbf{w} \mid \mathcal{D})$ is Gaussian, but we don't need to approximate the integral in that case. For logistic regression there's no obvious way to draw samples from the posterior distribution (if we don't approximate it with a Gaussian).

A family of methods, widely used in Statistics, known as *Markov chain Monte Carlo* (MCMC) methods, can be used to draw samples from the posterior distribution for models like logistic regression and neural networks. We don't cover the details of MCMC in this course. If you're interested, Iain has a tutorial here: `https://homepages.inf.ed.ac.uk/imurray2/teaching/15nips/` — or a longer tutorial on probabilistic modelling that puts it in slightly more context: `https://homepages.inf.ed.ac.uk/imurray2/teaching/14mlss/`

1.1 Importance Sampling

Importance sampling is a simple trick you should understand, because it comes up in various contexts in machine learning beyond Bayesian prediction[1]. Here we rewrite the integral as an expectation under an arbitrary tractable distribution of our choice, $q(\mathbf{w})$:

$$P(y=1 \mid \mathbf{x}, \mathcal{D}) = \int \sigma(\mathbf{w}^\top \mathbf{x})\, p(\mathbf{w} \mid \mathcal{D})\, \frac{q(\mathbf{w})}{q(\mathbf{w})}\, \mathrm{d}\mathbf{w} \tag{4}$$

$$= \mathbb{E}_{q(\mathbf{w})} \left[\sigma(\mathbf{w}^\top \mathbf{x})\, \frac{p(\mathbf{w} \mid \mathcal{D})}{q(\mathbf{w})} \right] \tag{5}$$

$$\approx \frac{1}{S} \sum_{s=1}^{S} \sigma(\mathbf{w}^{(s)\top} \mathbf{x})\, \frac{p(\mathbf{w}^{(s)} \mid \mathcal{D})}{q(\mathbf{w}^{(s)})}, \quad \mathbf{w}^{(s)} \sim q(\mathbf{w}). \tag{6}$$

Here $r^{(s)} = \frac{p(\mathbf{w}^{(s)} \mid \mathcal{D})}{q(\mathbf{w}^{(s)})}$ is the *importance weight*, which upweights the predictions for parameters that are more probable under the posterior than the distribution we sampled from.

1. For example reweighting data in a loss function to reflect how they were gathered, or weighting the importance of different trial runs in reinforcement learning, depending on the policy from which they were sampled.

We shouldn't divide by zero, so we need $q(\mathbf{w}) > 0$ when $p(\mathbf{w} \mid \mathcal{D}) > 0$. Moreover, we don't want $q(\mathbf{w}) \ll p(\mathbf{w} \mid \mathcal{D})$ for any region of the weights, or occasionally we would see an enormous importance weight, and the estimator will have high variance.

The final detail is that we can't usually evaluate the posterior

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \mathbf{w})\, p(\mathbf{w})}{P(\mathcal{D})}, \tag{7}$$

because we can't usually evaluate the denominator $p(\mathcal{D})$. However, we can approximate that using importance sampling!

$$\begin{aligned} P(\mathcal{D}) &= \int P(\mathcal{D} \mid \mathbf{w})\, p(\mathbf{w})\, \mathrm{d}\mathbf{w} && (8)\\ &= \int P(\mathcal{D} \mid \mathbf{w})\, p(\mathbf{w})\, \frac{q(\mathbf{w})}{q(\mathbf{w})}\, \mathrm{d}\mathbf{w} && (9)\\ &= \mathbb{E}_{q(\mathbf{w})} \left[\frac{P(\mathcal{D} \mid \mathbf{w})\, p(\mathbf{w})}{q(\mathbf{w})} \right] && (10)\\ &\approx \frac{1}{S} \sum_{s=1}^{S} \frac{P(\mathcal{D} \mid \mathbf{w}^{(s)})\, p(\mathbf{w}^{(s)})}{q(\mathbf{w}^{(s)})} = \frac{1}{S} \sum_{s=1}^{S} \tilde{r}^{(s)}, && (11) \end{aligned}$$

where we've introduced "unnormalized importance weights", defined as:

$$\tilde{r}^{(s)} = \frac{P(\mathcal{D} \mid \mathbf{w}^{(s)})\, p(\mathbf{w}^{(s)})}{q(\mathbf{w}^{(s)})}. \tag{12}$$

Substituting in this approximation to the Bayesian prediction, we obtain:

$$P(y=1 \mid \mathbf{x}, \mathcal{D}) \approx \frac{1}{S} \sum_{s=1}^{S} \sigma({\mathbf{w}^{(s)}}^\top \mathbf{x})\, \frac{\tilde{r}^{(s)}}{\frac{1}{S} \sum_{s'=1}^{S} \tilde{r}^{(s')}}, \quad \mathbf{w}^{(s)} \sim q(\mathbf{w}) \tag{13}$$

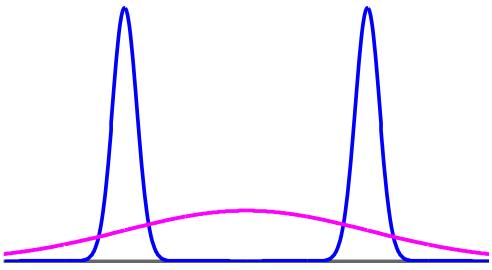
or

$$P(y=1 \mid \mathbf{x}, \mathcal{D}) \approx \sum_{s=1}^{S} \sigma({\mathbf{w}^{(s)}}^\top \mathbf{x})\, r^{(s)}, \quad \mathbf{w}^{(s)} \sim q(\mathbf{w}). \tag{14}$$

In this final form, the average is under the distribution defined by the 'normalized importance weights':

$$r^{(s)} = \frac{\tilde{r}^{(s)}}{\sum_{s'=1}^{S} \tilde{r}^{(s')}}. \tag{15}$$

Consider a 1-dimensional bimodal posterior $p(w \mid \mathcal{D})$ and $q(w)$ a Gaussian centred at the trough of $p(w \mid \mathcal{D})$ as shown in the figure below.



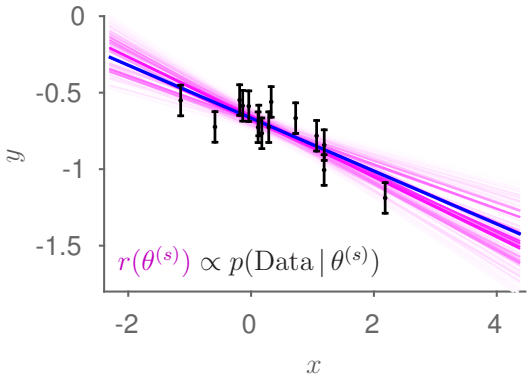
[The website version of this note has a question here.]

1.2 Importance sampling with the prior

A special case might help understand importance sampling. If we sampled model parameters from the prior, $q(\mathbf{w}) = p(\mathbf{w})$, the unnormalized weights are equal to the likelihood, $\tilde{r}(\mathbf{w}) = P(\mathcal{D} \mid \mathbf{w}^{(s)})$.

We would sample some number, S, settings of the parameters from the prior. Then we form a discrete distribution over these parameters with importance weights proportional to the likelihood. Functions that match the data will be given large importance weight.

Below is a linear regression example, where the true (unknown) line is shown in blue, and the purple lines show the discrete distribution over possible models we will use for prediction. The intensity of the lines indicate the importance weights. I drew 10,000 samples from the prior, but most of the functions didn't go near the data and were given such small weight that they are nearly white.



This importance sampling procedure works in principle for any model where we can sample possible models from the prior and evaluate the likelihood, including logistic regression. However, if we have many parameters, it is unlikely that *any* of our S samples from the prior will match the data well, and our estimates will be poor.

We could try to make the sampling distribution $q(\mathbf{w})$ approximate the posterior, but for models with many parameters it is difficult to approximate the posterior well enough for importance sampling to work well. Advanced sampling methods like MCMC (mentioned above) and more advanced importance sampling methods (e.g., *Sequential Monte Carlo*, SMC) have been applied to neural networks, but are beyond the scope of this course.