IAIN STYLES

LOGISTIC REGRESSION

Logistic Regression

In Naïve Bayes, we determine the decision boundary by explicitly modelling the distribution of each class. We showed that this can be an effective classification method, even if the assumptions of the underlying model (conditionally independent input variables, and Gaussian class-conditional probability density functions for numeric input variables) are not met. We will look at an alternative probabilistic approach to classification that does not rely on explicit modelling of the likelihood: *logistic regression*. As the name suggests, this is related to the regression problem for continuous variables that we studied earlier in the module.

We will formulate logistic regression initially for two-class binary classification. The key idea is that we will model the probability (actually the odds) of an observation being in a class as a linear combination of the independent variables. Given a set of input features, we will seek to return one of two possible output values $Y = \{0, 1\}$ that indicate which of the two classes a sample belongs to.

The key quantity that we will work with is *odds*, defined as the ratio of the probabilities of the two possible outcomes. If we denote $p_i(\mathbf{x}) = P(c_i|\mathbf{x})$, then the odds that the measurement is in class c_1 is $o_1 = p_1/p_0 = p_1/(1-p_1)$ since we have a two-class problem. The logarithm of the odds – the so-called *logit* – is then

$$\text{logit}(p_1) = \ln\left(\frac{p_1}{1-p_1}\right) \tag{1}$$

We might reasonably ask why this is a sensible thing to do. The posterior probabilities $P(c_i|\mathbf{x})$ are, of course, limited to the range $[0, 1]$. If we wish to formulate classification as a regression problem, we need to map the possible outputs of a regression model which lie in the range $(-\infty, \infty)$, onto the possible classification outputs so that we can build a regression model. This is what the logit function does, as shown in Figure 1. It maps $(0, 1) \mapsto (-\infty, \infty)$ with a sharp "transistion zone" between the extremes that effectively maps the continuous input into a binary output with a thin non-binary zone between the two extremes.

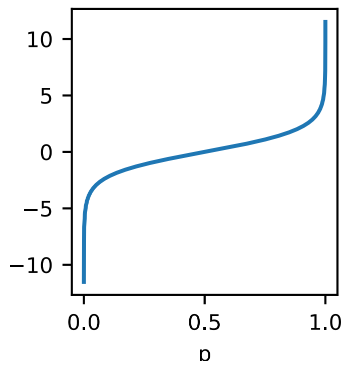

Figure 1: The logit function maps $[0, 1]$ to $[-\infty, \infty]$.

Now that we have transformed the probability to a continuous variable in $\mathbb{R}$, we can model it as a regression problem. We write

$$\text{logit}(p_1) = \ln\left(\frac{p_1}{1-p_1}\right) = w_0 + w_1x_1 + \cdots + w_nx_n = \mathbf{w}^\text{T}\mathbf{x} \tag{2}$$

where we use only the linear terms in the independent variables $\mathbf{x} = 1, x_1, x_2, \ldots, x_M$. Taking the exponential of both sides and rearranging, we can show that

$$p_1 = \frac{\exp(\mathbf{w}^\text{T}\mathbf{x})}{1+\exp(\mathbf{w}^\text{T}\mathbf{x})} \quad \text{and} \quad p_0 = 1 - p_1 = \frac{1}{1+\exp(\mathbf{w}^\text{T}\mathbf{x})} \tag{3}$$

The probability p_1 therefore depends on $\mathbf{x}$ and $\mathbf{w}$ so we will write this as $p_1(\mathbf{x}, \mathbf{w})$.

We now formulate the likelihood over a set of observations and maximise this to train the model. Given N independent observations $\{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$, the overall likelihood is the product of the likelihoods of the individual observations. Noting that the observations y_i are binary ($\{0, 1\}$), the joint likelihood can be written:

$$\mathcal{L}(\mathbf{w}) = \prod_{i=1}^{N} p_1(\mathbf{x}^{(i)}, \mathbf{w})^{y^{(i)}} p_0(\mathbf{x}^{(i)}, \mathbf{w})^{1-y^{(i)}} \tag{4}$$

$$= \prod_{i=1}^{N} p_1(\mathbf{x}^{(i)}, \mathbf{w})^{y^{(i)}} \left(1 - p_1(\mathbf{x}^{(i)}, \mathbf{w})\right)^{1-y^{(i)}} \tag{5}$$

As we did in the probabilistic formulation of regression, we choose $\mathbf{w}$ by maximising the likelihood, or equivalently the log-likelihood which is

$$\ln \mathcal{L}(\mathbf{w}) = \ln(\mathcal{L}(\mathbf{w})) = \sum_{i=1}^{N} y_i \ln(p_1(\mathbf{x}^{(i)}, \mathbf{w})) + (1 - y^{(i)}) \ln(1 - p_1(\mathbf{x}^{(i)}, \mathbf{w})) \tag{6}$$

$$= \sum_{i=1}^{N} y^{(i)} \left[\ln(p_1(\mathbf{x}^{(i)}, \mathbf{w})) - \ln(1 - p_1(\mathbf{x}^{(i)}, \mathbf{w}))\right] + \ln(1 - p_1(\mathbf{x}^{(i)}, \mathbf{w})) \tag{7}$$

$$= \sum_{i=1}^{N} y^{(i)} \ln \frac{p_1(\mathbf{x}^{(i)}, \mathbf{w}))}{1 - p_1(\mathbf{x}^{(i)}, \mathbf{w})} + \ln(1 - p_1(\mathbf{x}^{(i)}, \mathbf{w})) \tag{8}$$

$$= \sum_{i=1}^{N} y^{(i)} \mathbf{w}^{\mathrm{T}} \mathbf{x}^{(i)} - \ln(1 + \exp(\mathbf{w}^{\mathrm{T}} \mathbf{x}^{(i)})). \tag{9}$$

The optimal weights $\mathbf{w}^*$ that maximise the likelihood are therefore given by

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \left[\sum_{i=1}^{N} y^{(i)} \mathbf{w}^{\mathrm{T}} \mathbf{x}^{(i)} - \ln(1 + \exp(\mathbf{w}^{\mathrm{T}} \mathbf{x}^{(i)}))\right] \tag{10}$$

This is as far as we can get analytically, and this quantity can't, in general, be maximised by hand to provide a closed-form solution. The normal method for solving this is to use an iterative method of solution called IRLS – Iterative Reweighted Least Squares – but a detailed explanation of this is beyond the scope of this course. The interested reader should refer to Section 8.3 of Izenman[1].

[1] Izenman, Multivariate Statistical Analysis, Springer (New York) 2013

Given the optimal weight vector $\mathbf{w}^*$, we are then in a position to compute the decision boundary, with reference to Equation (2). A data point $\mathbf{x}$ should be assigned to class 1 if $p_1 > 1 - p_1$, which is when $\mathbf{logit}(p_1) > 0$. The decision rule is therefore

$$\mathbf{w}^{*\mathrm{T}} \mathbf{x} > 0 \quad \rightarrow \quad \mathbf{x} \in c_1$$

Alternatively, we can compute the probabilities directly, recalling that

$$p_1 = \frac{\exp(\mathbf{w}^{*\mathrm{T}} \mathbf{x})}{1 + \exp(\mathbf{w}^{*\mathrm{T}} \mathbf{x})} \quad \text{and} \quad p_0 = \frac{1}{1 + \exp(\mathbf{w}^{*\mathrm{T}} \mathbf{x})} \tag{11}$$

and then $\mathbf{x}$ is assigned to the class with the higher probability.

It is useful to draw some comparisons between logistic regression (LR) and Naïve Bayes. Both are formulated from a statistical perspective, but there are some key differences that have been written about in the literature.

1. In LR, there is no assumption that the likelihoods are are Gaussians. In principle, this makes LR more robust to non-normality than Naïve Bayes.
2. Naïve Bayes assumes conditional independence, whereas in LR the independent variables should not be highly linearly correlated.
3. LR is much less efficient than Naïve Bayes for large sample sizes.
4. LR can require larger dataset sizes to work well.

Multiclass Logistic Regression

The generalisation of LR to multiple (M) classes is straighforward. The logit $L_i = \ln(\frac{p_i}{1-p_i}) = \mathbf{w}_i^{*\mathrm{T}}\mathbf{x}$ is computed for every class boundary i, and the class with the highest probability is selected. One common way to do this is to choose a single reference class as a "pivot" and compute all boundaries against that class, choosing the one with the highest probability. Given M classes, we compute the logit of every other class against one class. We will choose the "last" class M (noting that they can be reordered arbitrarily), and compute the logit as $\ln(\frac{p_i}{p_M})$ for all i.

$$\ln\frac{p_1}{p_M} = \mathbf{w}_1^{*\mathrm{T}}\mathbf{x} \tag{12}$$
$$\ln\frac{p_2}{p_M} = \mathbf{w}_2^{*\mathrm{T}}\mathbf{x} \tag{13}$$
$$\cdots \tag{14}$$
$$\ln\frac{p_{M-1}}{p_M} = \mathbf{w}_{M-1}^{*\mathrm{T}}\mathbf{x} \tag{15}$$
$$\tag{16}$$

Exponentiating, we have

$$p_i = p_M \exp(\mathbf{w}_i^{*\mathrm{T}}\mathbf{x}) \quad \text{for i=\{1,2,\ldots,M-1\}} \tag{17}$$

and because all probabilities must add to one we know that

$$\sum_{i=1}^{M} p_i = 1 \quad \rightarrow \quad p_M = 1 - \sum_{i=1}^{M-1} p_M \exp(\mathbf{w}_i^{*\mathrm{T}}\mathbf{x}) \tag{18}$$

and therefore

$$p_M = \frac{1}{1+\sum_{i=1}^{M-1}\exp(\mathbf{w}_i^{*\mathrm{T}}\mathbf{x})}. \tag{19}$$

Finally, we substitute this into Equation (17) to obtain

$$p_i = p_M \exp(\mathbf{w}_i^{*\mathrm{T}}\mathbf{x}) = \frac{\exp(\mathbf{w}_i^{*\mathrm{T}}\mathbf{x})}{1+\sum_{i=1}^{M-1}\exp(\mathbf{w}_i^{*\mathrm{T}}\mathbf{x})} \tag{20}$$

Thus we perform multiple binary LRs of each class against the pivot class c_M to find the parameters $\mathbf{w}_i^*$ for each class i, and assign $\mathbf{x}$ to the class with the highest probability p_i. As before, these regressions cannot be solved in closed-form and we must do this numerically with IRLS.

LR is implemented in most machine learning libraries. In the accompanying Jupyer Notebook, we use the version implemented in `scikit-learn` to run LR on the MNIST dataset. This can be found at `https://colab.research.google.com/drive/1-vpNgx3PtdyRGv1pC0PYrR-X-vdf8jJT`.

Reading

Section 4.3 of Bishop, Pattern Recognition and Machine Learning.