Inf2 – Foundations of Data Science 2022

Topic: Logistic regression

FOUNDATIONS OF DATA SCIENCE

14th January 2023

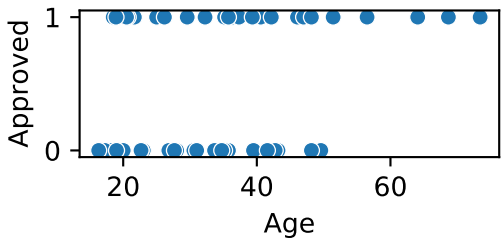
1 Principle of logistic regression

What is logistic regression used for? Logistic regression has various uses, including:

- **As a parametric supervised classification algorithm.** For example, a bank has data on previous customers it has considered offering credit cards to, including predictor variables (independent variables) such as their age, income, housing status and employment status. Each of these sets of variables is labelled with the dependent variable of whether the credit card was approved. The task is to determine if the credit card should be approved for a new customer.
- **As a way of investigating the association between predictors/independent variables and a binary (also called dichotomous) response variable.** For example, suppose we have an observational study of patients of different ages, health levels, ethnicity and gender. Some of the patients have had a dose of vaccine for an illness, and some haven't. We'd like to know how the probability of getting the illness depends on if the vaccine has been administered or not. Like multiple linear regression, we can examine logistic regression coefficients to isolate the effect of the vaccine, controlling for the other variables.

Similarities and differences to k-NN We've already discussed classifiers, when we looked at k-Nearest Neighbours in the topic Intro to supervised learning: training, testing and validation: Nearest neighbours. As a reminder, the problem of classification is to predict the correct label for an unlabelled input item described by a feature vector of variables. As well as acting as a classifier, logistic regression can predict a real-valued number, the *probability* of a data point belonging to a category, on the basis of the predictors/independent variables. In fact, we convert the logistic regression model into a classifier by choosing at a threshold level of probability at which we make a decision. For example, we might only want to approve cards that we think would have a 60% chance of being approved historically.

Association between continuous predictor and binary outcome We will use the example of the credit card approval to illustrate how logistic regression is used as a classifier and as a way of exploring associations between variables. Figure 1 visualises the relationships between age and approval and between employment status and approval in two ways. Because age is a continuous variable, we can plot individual datapoints on a scatter plot (Figure 1a). It looks like older customers were more likely to have their credit approved than younger ones.



(a) Age versus approval. Each datapoint represents the age of a customer and whether their credit was approved (1) or not approved (0). It looks like a greater fraction of older customers had their credit approved than younger ones. A random subsample of data points is plotted to aid visualisation.

Employed	Approved	Not approved	Approval odds
0	0.25	0.75	0.34
1	0.71	0.29	2.42

(b) Contingency table showing empirical probabilities of approval ("Success") or not approval ("Failure") based on employment status (first two columns) and the odds of approval (final column). The odds are the probability of approval divided by the probability of not being approved.

Figure 1: Relationship between age, employment status and credit approval in the credit approval dataset.

Association between binary predictor and binary outcome: Odds and odds ratios Employment is a binary variable ("employed" or "not employed"). If we tried plotting it in the same way as age versus approval, we'd end up with a very uninformative plot, so instead we look at a **contingency table** (Figure 1b), which shows the empirical probability (relative frequency) of having credit approved or not approved based on employment status.

In logistic regression, we will see that it makes sense to describe these probabilities in terms of **odds**[1], which we define as:

$$\text{Odds}(\text{Success}) = \frac{P(\text{Success})}{P(\text{Failure})} = \frac{P(\text{Success})}{1 - P(\text{Success})} \tag{1}$$

If success and failure are equally likely, the odds are equal to 1.

We call the **odds ratio** (OR) the ratio between the odds of credit approval if employed versus credit approval if not employed.

$$\text{OR}(x) = \frac{\text{Odds}(\text{Success}|x = \text{True})}{\text{Odds}(\text{Success}|x = \text{False})} \tag{2}$$

We can find the odds ratio of employment in the credit example by setting "Success" to "Approved" and x to "Employed", giving an answer of $7.09 = 2.42/0.34$. Thus, the odds of

[1] *Modern Mathematical Statistics with Applications* calls the "odds" the "odds ratio", which is not standard usage.

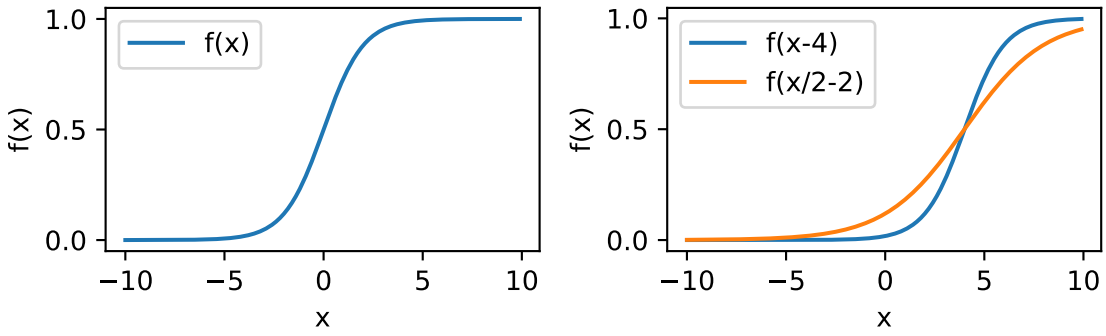


Figure 2: The logistic function. Left: the standard logistic function: $f(u) = \exp(u)/(1+\exp(u))$, which can also be written $f(u) = 1/(1+\exp(-u))$. Right: Examples of shifted logistic curves.

someone who is employed having credit approved are 7.09 times larger than the odds of someone who is not employed having credit approved. The odds ratio is sometimes referred to as an **effect size** and expressed as the percentage change in the odds from x being False to True; this case the effect size is 609%, since the effect of employment increases the odds of approval by this amount.

Principle of logistic regression with one dependent variable As its name suggests, logistic regression is related to linear regression. Suppose that the dependent (or response) variable y is a dichotomous variable (i.e. a categorical variable with two categories). We'll represent the categories by 0 (failure) and 1 (success). We'd like to model the probability $P(Y = 1|X = x)$ that the response variable is 1, given the independent variable (or predictor) X has a value x. Because we're predicting a probability, the answer given by logistic regression has to lie between 0 and 1. Therefore, $P(Y = 1|X = x)$ can't be a linear function of x.

We get around this problem by allowing $P(Y = 1|X = x)$ to be a *nonlinear* function of x. A function that works well in many applications is the **logistic function**[2] (Figure 2). Using f to denote the logistic function, the probability of a success is:

$$P(Y = 1|X = x) = f(\beta_0 + \beta_1 x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}} = \frac{1}{1 + e^{-\beta_0-\beta_1 x}} \tag{3}$$

Just as with linear regression, we can adjust the values of the coefficients β_0 and β_1 to fit the data as best as possible – we will come to how we do this later.

[2]Not-examinable: The logistic function is also known as the sigmoid function, and denoted $S(x)$ or $\sigma(x)$, due to its S-shaped curve. However, the term "sigmoid function" can refer to a family of S-shaped functions.

The term *logisitique* was first used in 1845 by Verhulst (1845) to describe the solution of a differential equation describing population growth: $\frac{dp}{dt} = p(1-p)$. However, Verhulst applied the term *logisitique* to the expression of time in terms of population, i.e. essentially $t = \ln(p/(1-p))$. This is in fact the "logit" function: $\text{logit}(p) = \ln(p/(1-p))$, which is the *inverse* of the logistic function. The name logit was coined much later – see later footnote.

In python `scipy` and some R packages, the logistic function is referred to as "expit", making "expit" the inverse of "logit", just as "log" is the inverse of "exp".

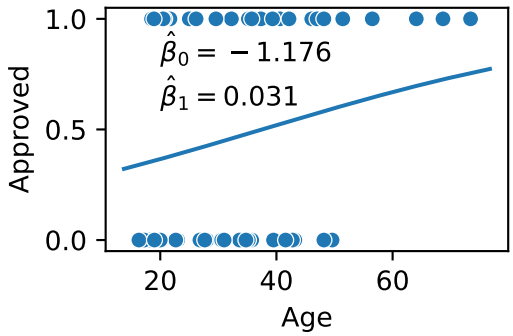


Figure 3: Logistic regression of age on credit approval.

Application to credit example with one variable Figure 3 shows the logistic regression of age on credit approval – we are ignoring all the other variables for now. The curve doesn't look very much like a logistic curve, but that's because it's got a very shallow slope, since $\hat{\beta}_1 = 0.03$. We can see that the probability ranges between about 0.37 for teenagers and 0.8 for 70-year-olds.

2 Interpretation of logistic regression coefficients

Interpretation of $\hat{\beta}_0$ In linear regression $\hat{\beta}_0$ is the intercept: it tells us the predicted value of the dependent variable when the independent variable is 0. In logistic regression $f(\hat{\beta}_0) = 1/(1 + \exp(\hat{\beta}_0))$ tells us the probability the dependent variable being 1 ("success") when the independent variable is 0. In the credit example it suggests the likelihood of a newborn baby receiving credit approval is $f(-1.176) = 0.236$ – which seems rather high!

Log odds Remember the definition of odds (Equation 1). To interpret the coefficient $\hat{\beta}_1$ it helps to rewrite the logistic regression model (Equation 3) in terms of **log odds**, i.e. the log of the odds:

$$\text{Log Odds(Success)} = \ln \frac{P(\text{Success})}{P(\text{Failure})} = \ln \frac{P(\text{Success})}{1 - P(\text{Success})} \tag{4}$$

Log odds of 0 mean that success and failure are equally likely: $P(\text{Success}) = P(\text{Failure}) = 0.5$. Positive log odds mean that success is more likely than failure, and vice versa for negative log odds. An increase of 1 unit of the log odds means that the odds increase by a factor of e. As the probability tends towards 1, the log odds tend towards infinity; as the probability tends towards 0, the log odds tend towards negative infinity.

When we express probability in terms of log odds, we sometimes say it has units of "logits", which stands for *log*istic un*it*s. Going back to the example, we can say that when the independent variable is 0, the log odds of approval are $\hat{\beta}_0 = -1.176$ logits.

The logit function converts the probability of success into the log odds of success to failure[3]:

$$\text{logit}(p) = \ln \frac{p}{1-p} \tag{5}$$

We can now re-express Equation 4 as Log Odds(Success) = logit(P(Success)).

Rewriting the logistic regression model in terms of log odds The probability of a failure is:

$$P(Y=0|X=x) = 1 - f(\beta_0 + \beta_1 x) = 1 - \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} = \frac{1}{1+e^{\beta_0+\beta_1 x}} = f(-\beta_0 - \beta_1 x) \tag{6}$$

We can divide Equation 3 by Equation 6 to obtain[4]:

$$\frac{P(Y=1|X=x)}{P(Y=0|X=x)} = \frac{P(Y=1|X=x)}{1-P(Y=1|X=x)} = e^{\beta_0+\beta_1 x} \tag{7}$$

The ratio on the left is the odds for success. It tells us how many times more likely the "success" ($Y=1$) is than the "failure" ($Y=0$) for any value of x (see Equation 1). If we take natural logs of both sides of the equation, we see that the log odds is a linear function of the predictor:

$$\text{logit}(P(Y=1|X=x)) = \ln \frac{P(Y=1|X=x)}{1-P(Y=1|X=x)} = \ln \frac{P(Y=1|X=x)}{P(Y=0|X=x)} = \beta_0 + \beta_1 x$$

We can now see that $\hat{\beta}_0$ is the log odds when the independent variable is equal to 0.

Interpretation of $\hat{\beta}_1$ From Equation 2, we can see that the parameter β_1 tells us the increase in the log odds when we increase x by 1 unit.

In other words, when we increase x by 1 the odds multiply by a factor $\exp(\beta_1)$. We refer to this factor as the odds ratio (OR) for the variable x. In this example the OR = $\exp(0.03) = 1.03$. Thus, for every year of age, you're 1.03 times more likely to have a loan approved, an effect size of 3%.

3 Multiple logistic regression and confidence intervals

Principle of multiple logistic regression Just as with multiple regression, we can extend the logistic regression model (Equation 3) to include extra independent variables to with

[3] If we have a continuous dependent variable between 0 and 1 (e.g. the proportion p of organisms killed by a toxin), we could transform the dependent variable into logits using logit(p). In fact, logistic regression and the term logit were invented to deal with this sort of data (Berkson, 1944).

[4] This identity should help to see this:

$$\frac{f(u)}{f(-u)} = \frac{e^u}{1+e^u}\frac{1+e^{-u}}{e^{-u}} = e^u$$

	Variable	Coefficient	Odds or OR
$\hat{\beta}_0$	Intercept	-1.969	0.140
$\hat{\beta}_1$	Age	0.029	1.030
$\hat{\beta}_2$	Employed	1.881	6.562

Table 1: Coefficients expressed in raw form and as odds ratio $\exp(\beta)$.

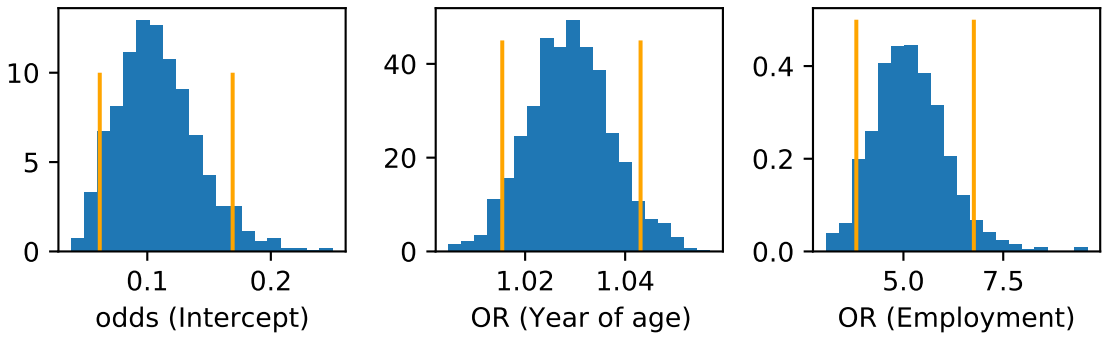


Figure 4: Bootstrap distributions for the baseline odds and odds ratios for age and employment in the credit scoring example.

corresponding coefficients:

$$P(Y = 1|X^{(1)} = x^{(1)}, X^{(2)} = x^{(2)}, \dots) = f(\beta_0 + \beta_1 x^{(1)} + \beta_2 x^{(2)} + \dots) \tag{8}$$

If the independent variables are binary (such as employment status) we can just include them as binary variables.

Multiple logistic regression applied to credit example If we apply multiple logistic regression to the credit example, we end up with the coefficients and odds ratios shown in Table 1. We can see that the effect of being employed increases the odds of being awarded a loan by a factor of 6.56, an effect size of 556%. By contrast each year of employment only multiplies the odds by 1.03, an effect size of 3%. To see the effect of increasing age by 10 years, we'd need to raise this OR to the power 10, and would find that the odds are only multiplied by 1.35. The effect an increase in age from 20 to 70 is about 4.36 – still less than the effect of being in employment.

Bootstrap confidence intervals on coefficients Just as the mean and median are statistics, so are the coefficients $\hat{\beta}_1$ and $\hat{\beta}_2$ in logistic regression. We can therefore use the bootstrap to generate confidence intervals for logistic regression (Figure 4). The central estimate and the 95% confidence intervals computed from the 2.5% and 97.5% centiles are:

- Age: OR=1.030, CI=(1.017, 1.044)
- Employment: OR=6.562, CI=(5.110, 8.805)

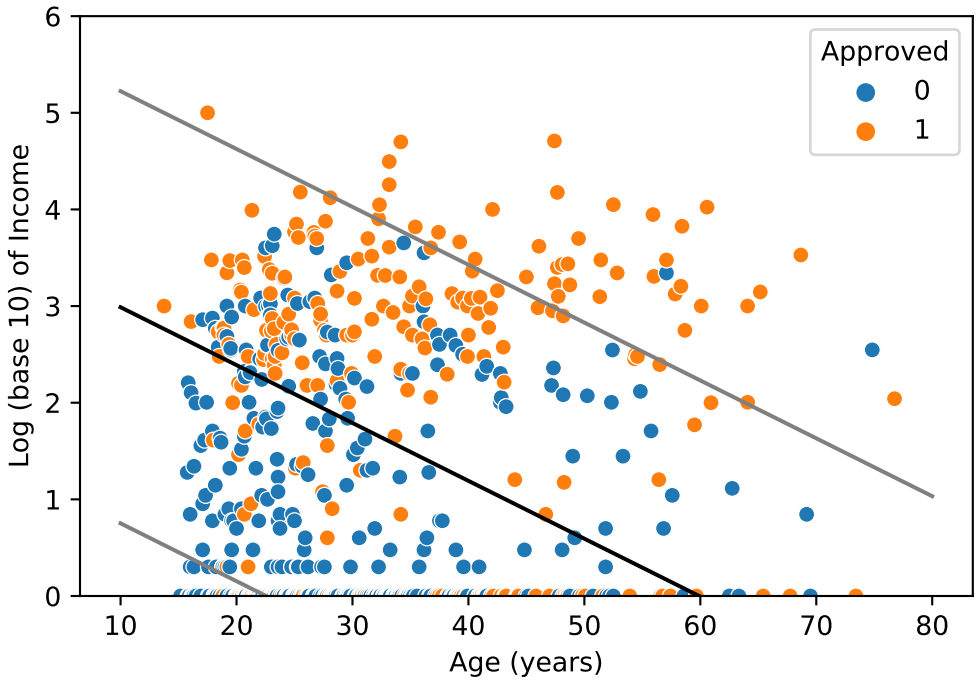


Figure 5: Logistic regression applied to credit approval dataset. The age and log income of each application is plotted, along with its approval status. The black line is the decision boundary found by logistic regression with an odds ratio of 1, i.e. log odds $c = 0$. The grey lines are the thresholds corresponding to odds ratios of 3 and 1/3 (i.e. 75%/25% and 25%/75%.

4 Logistic regression as a classifier

Converting linear regression to a classifier Setting a threshold probability p_{thresh} corresponds to setting threshold log odds, which we'll define as c. If we choose log odds $c = 0$, this means odds of 1, i.e. the probability of success (approval) is 1/2. Substituting $c = \ln \frac{P(Y=1|x)}{1-P(Y=1|x)}$ into Equation 2, we find:

$$c = \beta_0 + \beta_1 x \tag{9}$$

This defines a linear decision boundary – in the region where $\beta_0 + \beta_1 x > c$, the log odds are greater than the threshold, and we classify unseen datapoints in this region as "Success", and elsewhere, we classify unseen datapoints as "Failure".

Figure 5 shows decision boundaries for various threshold levels when we consider two continuous variables in the credit data set: age and the log of the income. Note: as with linear regression, it often makes sense with logistic regression to transform variables so that their distribution is as normal as possible.

Transparency of logistic regression The credit agency might want to explain to its customers why their application was or was not approved. Logistic regression makes it very easy to do

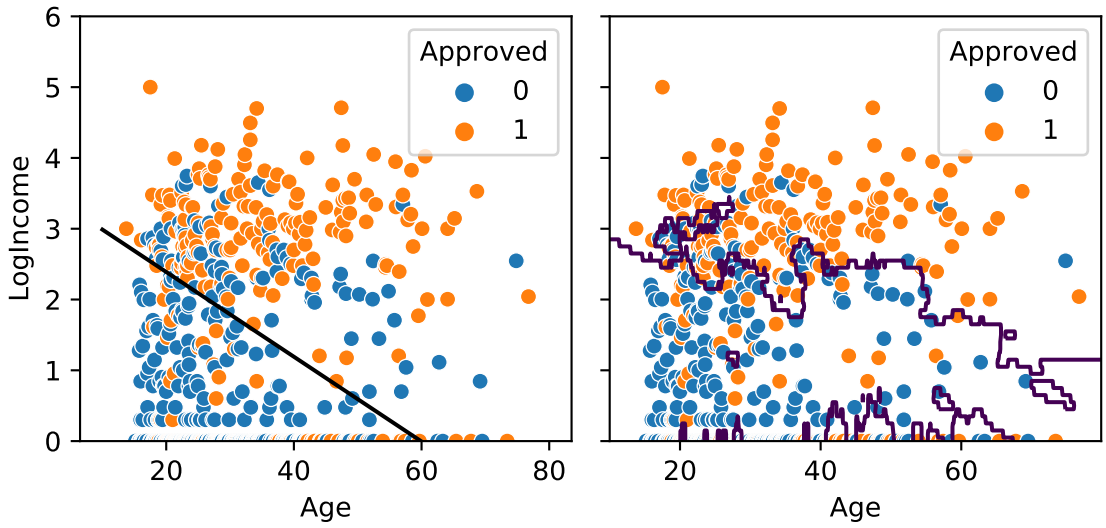


Figure 6: Logistic regression (left) versus 11-NN (right) applied to the credit data. The decision boundaries are shown as dark lines.

this, since essentially we have a credit scoring system:

- If you are in employment you score 1.625, if not you score 0
- Multiply your age by 0.029 and add the result to your score
- Round your income to the nearest 1000. Multiply the number of zeros in this figure by 0.320 and add the result to your score[5]
- If you scored more than 2.246, your credit will be approved

Thus, a logistic regression classifier is potentially a very **transparent** classifier. It could help to reduce the ethical harms of data science to individuals by allowing them to understand why their loan was rejected. One of the recommendations of Vallor's *Introduction to Data Ethics* is to "Promote Values of Transparency, Autonomy, and Trustworthiness" (Vallor, 2018).

Logistic regression versus *k*-nearest neighbour The logistic regression classifier differs in a number of ways from the nearest neighbour classifier:

1. The logistic regression decision boundary (obtained by setting a probability criterion) is a straight line, whereas the nearest neighbour decision boundary is nonlinear (Figure 6).
2. The k-NN thus gives more flexibility and the ability to have higher accuracy, but it is also more likely to over-fit – remember the topic on E valuation.

[5]OK, this is an approximation to a log! We could ask people to take logs or provide a table: or make the algorithm itself work in this way.

3. The logistic regression algorithm is more transparent than k-NN.

4. k-NN classifiers benefit from having standardised independent variables as inputs; logistic regression doesn't need this, though it can help if we are regularising a logistic regression classifier (which we will not do in this course).

Often it is worth trying logistic regression first in classification problem.

5 Maximum likelihood estimation of logistic regression coefficients

Principle of maximum likelihood We now turn to how to estimate the logistic regression coefficients to give the best estimates $\hat{\beta}_0$, $\hat{\beta}_1$ etc. In *linear* regression we minimised the sum of squared errors between the predicted and observed dependent variables. We could do this analytically, ending up with a formula for the regression coefficients. In logistic regression, it doesn't make sense to minimise the sum of squared errors, since our dependent variable is only 0 or +1 whereas the independent variables can have an infinite range.

Instead, we use the **principle of maximum likelihood**, which states that we adjust the model coefficients so as to maximise the likelihood that the observed data arises from the model. The resulting coefficients are referred to as the maximum likelihood estimators. Maximum likelihood can be applied to many models, not just logistic regression.

To apply the principle of maximum likelihood, we need an expression for the likelihood of all the observed data given the model, which we will do below. The likelihood is a function of β_0 and β_1. However, unlike in the case of linear regression, we can't derive formulae to give the best estimates of the coefficients $\hat{\beta}_0$ and $\hat{\beta}_1$ that maximise it. We have instead to use a numerical optimisation procedure that gets us to the best estimates.

Intuition of maximum likelihood applied to logistic regression The maximum likelihood principle and derivation may look a bit complicated. We can imagine that the logistic function $P(Y = +1|X = x_i)$ is like a piece of elastic. Datapoints that are "successes" ($y_i = 1$) pull $P(Y = 1|X = x_i)$ upwards towards 1 at the location x_i, since this will make success more likely. Datapoints that are "failures" ($y_i = 0$) pull $P(Y = 1|X = x_i)$ downwards towards 0 at the location x_i. Of course, the successes and failures may be mixed up, in which case they will be competing with each other to pull the logistic function up or down.

Derivation of maximum likelihood function Assuming that all our datapoints are independent of each other, the likelihood of obtaining the full set of response variables $y_1, \ldots, y_n$ is the product of the likelihood of getting each individual variable[6]. The probability of getting a

[6]We will use the product notation (Greek capital letter "pi" Π) to represent a product of probabilities. For example to the probabilities of three independents events happening is $p_1 p_2 p_3$, which we represent $\prod_{i=1}^{3} p_i$.

success or failure is given by Equations 3 and 6. We use a trick to combine them so that the probability of success used when $y_i = 1$ and the probability of failure is used when $y_i = 0$:

$$\prod_{i=1}^{n} P(Y = y_i|X = x_i) = \prod_{i=1}^{n} (y_i f(\beta_0 + \beta_1 x_i) + (1 - y_i) f(-\beta_0 - \beta_1 x_i)) \tag{10}$$

Substitute $y_i = 1$ or $y_i = 0$ in the equation above to verify that each one "selects" the correct probability.

We can now use another trick. Notice that $2y_i - 1$ is equal to 1 when $y_i = 1$ and equal to -1 when $y_i = 0$. We can now express the arguments of the logistic functions in terms of $2y_i - 1$:

$$\prod_{i=1}^{n} P(Y = y_i|X = x_i) = \prod_{i=1}^{n} (y_i f((2y_i - 1)(\beta_0 + \beta_1 x_i)) + (1 - y_i) f((2y_i - 1)(\beta_0 + \beta_1 x_i)))) \tag{11}$$

There's now a common factor, $f((2y_i - 1)(\beta_0 + \beta_1 x_i))$, so the equation simplifies:

$$\begin{aligned} \prod_{i=1}^{n} P(Y = y_i|X = x_i) &= \prod_{i=1}^{n} (y_i + 1 - y_i) f((2y_i - 1)(\beta_0 + \beta_1 x_i)) \\ &= \prod_{i=1}^{n} f((2y_i - 1)(\beta_0 + \beta_1 x_i)) \end{aligned} \tag{12}$$

Since the log function is a monotonically increasing function, maximising this probability is equivalent to maximising the log likelihood[7]:

$$\sum_{i=1}^{n} \ln P(Y = y_i|X = x_i) = \sum_{i=1}^{n} \ln f((2y_i - 1)(\beta_0 + \beta_1 x_i)) \tag{13}$$

The log of $f(u)$ is $-\ln(1 + e^{-u})$, so we can now write the optimisation equation as maximising:

$$\sum_{i=1}^{n} \ln P(Y = y_i|X = x_i) = \sum_{i=1}^{n} -\ln(1 + \exp(-(2y_i - 1)(\beta_0 + \beta_1 x_i))) \tag{14}$$

with respect to β_0 and β_1. Although we can compute the gradients with respect to β_0 and β_1, we can't solve the resulting equations analytically; we have to use numerical optimisation to find the best estimates $\hat{\beta}_0$ and $\hat{\beta}_1$.

References

Berkson, J. (1944). 'Application of the logistic function to bio-assay'. *Journal of the American Statistical Association* **39**:357–365

[7] The log law $\log ab = \log a + \log b$ can be generalised using product and sum notation: $\log\left(\prod_{i=1}^{n} x_i\right) = \sum_{i=1}^{n} \log x_i$.

Vallor, S. (2018). 'An introduction to data ethics'. Online. URL https://www.scu.edu/media/ethics-center/technology-ethics/IntroToDataEthics.pdf

Verhulst, P.-F. (1845). 'Recherches mathématiques sur la loi d'accroissement de la population'. *Nouveaux mémoires de l'Académie Royale des Sciences et Belles-Lettres de Bruxelles* **18**:4–55. URL https://gdz.sub.uni-goettingen.de/id/PPN129323640_0018?tify={%22pages%22:[14],%22panX%22:0.459,%22panY%22:0.815,%22view%22:%22info%22,%22zoom%22:0.721}