Math Textbook Template

Author Name

February 16th, 2020

Contents

1 Logistic Regression

Logistic Regression is a classification method that can be used to model the probability that some outcome Y belongs to a particular category. For example, a bank wishes to calculate the probability that a client will default (Y prediction) on his/her loan given information on the client's current financial situation, such as yearly income (input X).

In this scenario, the possible outcomes are binary (two-level response), either the client defaults or doesn't default over the life of the loan. Instead of trying to model the response variable Y directly, we can utilize logistic regression to model the *probability* that Y belongs to either category (default, no default).

The question we are asking in this situation (What is the probability that the client will default based on their income?) can be formulated as such:

$$\Pr(\textbf{default} = \text{Yes}|\text{Income}) \tag{1.0.1}$$

However in practice, a binary outcome such as yes/no can be represented by 1 or 0 instead (Yes = 1, No = 0) and our problem becomes:

$$\Pr(Y = 1|X) \tag{1.0.2}$$

Where X is the random variable representing income.

In the case of 2-level classification, one can use linear regression to loosely determine probabilities.

Let $p(X) = \Pr(Y = 1|X)$, then the linear regression model is:

$$p(X) = \beta_0 + \beta_1 X \tag{1.0.3}$$

However, the linear regression model output has a range below 0 and above 1, which makes no sense in the context of probability.

Since we are trying to model the probability that the response variable falls within a certain category, we need a function whos output is bounded between 0 and 1. That is:

$$p(X) \in [0, 1] \tag{1.0.4}$$

Enter the Sigmoid Function.

1.1 Sigmoid Function

The Sigmoid function $S(x)$ has a distinct 'S' shaped curve - also known as the **sigmoid curve** - which returns a value betwen 0 and 1. Generally it is a differentiable, monotonically increasing function that is defined for all $x \in \mathbb{R}$

There are many functions which satisfy these characteristics but the most commonly known sigmoid function is the **logistic function** defined by the formula:

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \tag{1.1.1}$$

And it looks like:

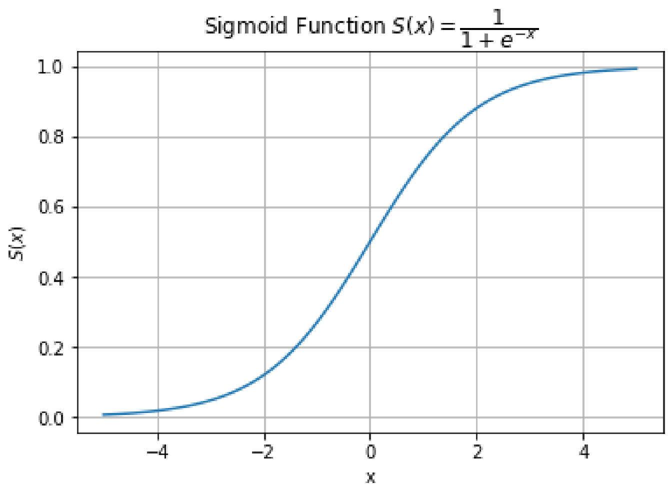

Figure 1: The Logistic Function - A Popular Sigmoid Function

Because of these properties ($S(x) \in (0, 1)$, monotonic increasing, differentiable), sigmoid functions are often used as **activation functions** for artificial neurons and are also used to as **cumulative distribution functions** in probability theory.

1.2 Logistic Regression Cont.

We left off on how linear regression can be used for classification, but since the output of the linear model can be above 1 or below 0, we a combination of the linear model and the logistic function:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} \tag{1.2.1}$$

Where β_0 and β_1 are coefficients to be determined, typically by *maximum likelihood* (Fitting the model).

By manipulating (1.2.1), we can obtain:

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X} \tag{1.2.2}$$

Where the left hand side of (1.2.2) is called the *odds* and can take on any value greater than zero. Odds close to zero indicate that the probability of the event occuring (in our case, default on a loan) is low. For example if $p(X) = 0.2$ - meaning probability of default is 20% - this implies that the odds are $\frac{0.2}{1-0.2} = \frac{1}{4}$.

On the other hand, high odds imply that the event in question is likely to occur. For example if $p(X) = 0.9$, then the implied odds become $\frac{0.9}{1-0.9} = 9$.

By taking the log of both sides of (1.2.2), the model becomes:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X \tag{1.2.3}$$

where the left-hand side of (1.2.3) is called the *log-likelihood* or *logit.* In this model, increasing X by one unit changes the log odds by β_1. Observe that if β_1 is positive, then an increasing X would be associated with an increasing $p(X)$. Likewise, if β_1 is negative, then an increasing X would be associated with a decreasing $p(X)$.

1.3 Logistic Regression - Estimating coefficients

As previously stated, the coefficients β_0 and β_1 are unknown and must be estimated by the training data.

The basic intuition behind maximum likelihood to fit the logistic regression model is to find estimates for β_0 and β_1 such that the predicted probability $\hat{p}(x_i)$ of default corresponds closely to the observed default status. In other words, we want to find β_0 and β_1 such that the output is close to 1 for all individuals who did default, and the output is cloes to 0 for all individuals who did not.

Maximum likelihood is used to estimate the unknown parameters β_0 and β_1 in (1.2.1).

Let $\beta = \{\beta_0, \beta_1\}$. In a two-class case the log-likelihood (log probability) can be written:

$$\ell(\beta) = \sum_{i=1}^{N} \left\{ y_i \log p(x_i;\beta) + (1-y_i)\log(1-p(x_i;\beta)) \right\} \tag{1.3.1}$$

where

$$p(x_i;\beta) = \Pr(G=1|X=x_i;\beta) \tag{1.3.2}$$

and x_i is the datapoint and β is the vector of two parameters in this case.

One way to maximize the log-likelihood is to set the derivative of (1.3.1) to zero and solve the *score equations*:

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i(y_i - p(x_i;\beta)) = 0 \tag{1.3.3}$$

To solve the score equation (1.3.3), we use the Newton-Raphson algorithm which is used to find roots of equations and requires the Hessian matrix (second derivative matrix):

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{N} x_i x_i^T p(x_i;\beta)(1-p(x_i;\beta)) \tag{1.3.4}$$

Each step of the iteration in the algorithm is defined:

$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 \ell(\beta^{old})}{\partial \beta \partial \beta^T}\right)^{-1} \frac{\partial \ell(\beta^{old})}{\partial \beta} \tag{1.3.5}$$

At this stage, its convenient to write the score equation and Hessian matrix in matrix notation.

Let $\mathbf{y}$ denote the vector of y_i values (vector of the dependent variable taking on values 1 or 0), $\mathbf{X}$ be the $N \times (p+1)$ matrix of x_i values (where each column of $\mathbf{X}$ is an input feature), $\mathbf{p}$ the vector of fitted probabilities with its i'th element $p(x_i; \beta^{old})$ and $\mathbf{W}$ an $N \times N$ diagonal matrix where the i'th diagonal element is $p(x_i; \beta^{old})(1 - p(x_i; \beta^{old}))$.

Now the derivatives become

$$\frac{\partial \ell(\beta)}{\partial \beta} = \mathbf{X}^T(\mathbf{y} - \mathbf{p}) \tag{1.3.6}$$

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -\mathbf{X}^T\mathbf{W}\mathbf{X} \tag{1.3.7}$$

Therefore each iteration for estimating β^{new} using the Newton-Raphson algorithm can be approximated by the following:

$$\begin{aligned} \beta^{new} &= \beta^{old} + (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T(\textbf{y-p}) \\ &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}(\mathbf{X}\beta^{old} + \mathbf{W}^{-1}(\mathbf{y} - \mathbf{p})) \\ &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{z} \end{aligned} \tag{1.3.8}$$

The second and third line of (1.3.8) re-expresses the Newton step as a weighted least squares step, with the response

$$\mathbf{z} = \mathbf{X}\beta^{old} + \mathbf{W}^{-1}(\mathbf{y} - \mathbf{p}) \tag{1.3.9}$$

also known as the *adjusted response.* With each iteration, (more on newton-raphson here)

2 Subset/Feature Selection

It is not uncommon to work on a data set with a large number of features, in which some parameters are highly correlated with others. This can lead to an *overfitted model*, which is a statistical model that contains more parameters than neccesary. It is possible that one can fit a model to a subset of the features that has stronger prediction power (lower predictive error) than a model fitted to the full subset of parameters.

In the case where the number of features is small, say 4 variables, its feasible to fit the model to all the possible combinations of variables which in this case, with $N = 4$, is

$$\text{Number of Combinations} = 2^4 - 1 = 15 \tag{2.0.1}$$

where N is the number of variables/parameters.

Since there are only 15 combinations of parameters, its computationally feasible to fit a model to all combinations to determine which subset of parameters will give you the strongest predictive power overall. However, imagine if you had 30 features in the data set. Then the number of combinations is:

$$\text{Number of Combinations} = 2^{30} - 1 = 1073741823 \tag{2.0.2}$$

With only 30 features, one must try over a billion combinations of subsets of features to determine the optimal subset. Therefore, brute force attempts at finding the optimal subset for fitting the model becomes impractical very quickly (as the number of features grows).

This section will introduce two popular shrinkage methods available for selecting subsets of predictors. Shrinkage methods fit a model with all n predictors using a technique that *constrains* or *regularizes* the coefficient estimates towards zero (e.g will shrink the coefficients of less important predictors to zero). Therefore, shrinkage methods can help one find the optimal subset of p features out of a total set of N features in the data set.

The two best-known shrinkage methods are Ridge Regression and Lasso Regression.

2.1 Ridge Regression

Ridge regression is similar to the least squares method to fit a linear model, where the sum of squared errors are being minimized to determine the coefficients. In ridge regression, the coefficients $\beta_0, \beta_1, \ldots, \beta_N$ are estimated by minimzing a penalty residual sum of squares (RSS)

$$\hat{\beta}^{\text{ridge}} = \text{RSS} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{2.1.1}$$

where $\lambda \geq 0$ is the *tuning parameter*, sometimes reffered to as the *complexity parameter*, which is to be determined separately. The term $\lambda \sum_{j=1}^{p} \beta_j^2$ is called the *shrinkage penalty*, which is small when $\beta_0, \beta_1, \ldots, \beta_p$ are close to zero and therefore has the effect of shrinking the estimates of β_j to towards zero. The tuning parameter λ acts as a scalar and controls the impact of the penalty term. When $\lambda = 0$, the penalty has no effect and ridge regression will produce a least squares estimate similar to OLS in linear regression. As $\lambda \to \infty$, the impact of the shrinkage penalty grows and the ridge regression coefficients will approach zero.

It is important to note that the ridge solutions are not equivariant under the scale of inputs, and one should standardize the inputs before solving (2.1.1), either through *Z-score scaling* or *min-max scaling.*

Equation (2.1.1) can be written

$$\hat{\beta}^{\text{ridge}} = \text{RSS}(\lambda) = (\mathbf{y} - \mathbf{X})\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^T\beta \tag{2.1.2}$$

Where the solution is

$$\hat{\beta}^{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{2.1.3}$$

Where $\mathbf{I}$ is the $p \times p$ identity matrix and input matrix $\mathbf{X}$ has p columns instead of $p + 1$ columns where the extra column is the $\mathbf{1}$ vector.

Picture to be removed

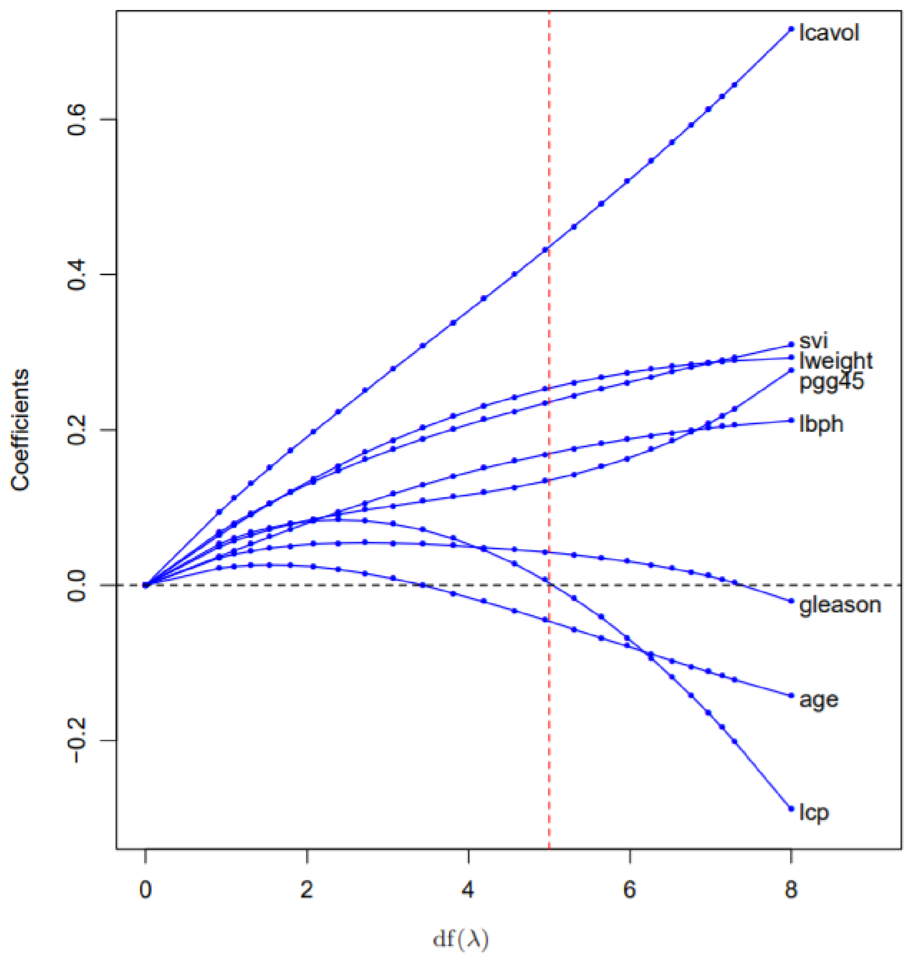

Figure 2: The Logistic Function - A Popular Sigmoid Function

2.2 Lasso Regression

Lasso Regression. Lasso Regression

2.3 Latex How to type a proof

$$\log \frac{\Pr(G = 1|X = x)}{\Pr(G = K|X = x)} = \beta_{10} + \beta_1^T x$$

$$\log \frac{\Pr(G = 2|X = x)}{\Pr(G = K|X = x)} = \beta_{20} + \beta_2^T x$$

References