Logistic Regression

(Course: Introduction to Data Science)

Tirtharaj Dash

BITS Pilani, K.K. Birla Goa Campus

tirtharaj@goa.bits-pilani.ac.in

October 24, 2020

Classification

- In the last lecture, we discussed linear regression. In this case, the output (y) is a real-value (possibly, in some range).
- Now, the question is: What if y is not real-value, rather a value from a discrete set. For example:
  - {dog, cat, elephant, monkey}
  - {car, bus, truck, bike}
  - {0, 1} or {-1, 1}
  - $\{0, 1, 2, \ldots, 999\}$
  - {win, loss, draw}

  Each element of a discrete set is called a 'class'.
- This type of problems in machine learning are called as classification problems.
- Goal is to predict a class label ($\hat{y}$) given a data instance represented by a feature vector ($\boldsymbol{x}$). Ideally, we would want that $\hat{y} = y$, where y is the true class label for $\boldsymbol{x}$.

Classification with Linear Regression? I

For simplicity, consider a two-class problem. If you want to use a linear regression model to solve this problem, here is what you may do:

- Represent one class as 'class 0' and other class as 'class 1'.
- Treat the class label as a real value. This is not wrong!
- Learn a linear regression model that treats these class labels are real-values and learn some weights for the features.
- In the end, you will get a linear regresion equation (with coefficients and intercept).

Is there a problem with this approach?

Classification with Linear Regression? II

Here is what can happen:

- Although, the class labels in your data are only $\{0, 1\}$, the linear regression model does not know that. It just treats those as two real numbers.
- So, there is absolutely no guarantee that the linear regression will produce either 0 or a 1 when asked to predict a class for a data instance.
- There is also a possibility that the outputs predicted by the linear regression is negative or greater than 1.
- Also, the linear regression model can be heavily affected by a few outliers in the data, making the regression line to be sensitive towards those points (if the amount of available data is small).

Classification with Linear Regression? III

To solve the stated issues in the previous slide, one straightforward option is to ask (actually, force) the linear regression model to produce a real number between 0 and 1.

Will this solve our problem? YES. How?

Assuming each class labels are equally sensitive, we can safely say that if the output real number is ≥ 0.5, the class label for the given data is 1; otherwise the class is 0.

Classification with Linear Regression? IV

Now, when we are restricting the output to be between 0 and 1, we can naturally think of a term called probability. The probability value ranges between $[0, 1]$.

Then, we can think of a function that has the same range. So, our goal now is to model probabilities.

The probability here is to be read as: The probability of the given data belonging to class 1 (or class 0*), given the feature representation of the data instance.

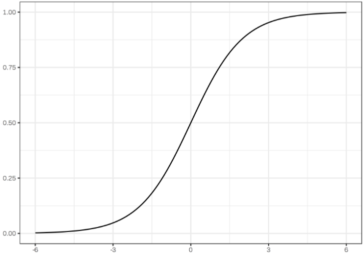
*Doesn't matter. This is representation dependent.

Modelling Probabilities I

- Instead of fitting a straight line (or hyperplane for a multivariate case), we can use the logistic function to squeeze the output of a linear equation between 0 and 1. The logistic function is defined as:

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

The function looks like this:



X-axis: z, Y-axis: $\sigma(z)$. When $z \to \infty, \sigma(z) \to 1$ and when $z \to -\infty, \sigma(z) \to 0$.

Modelling Probabilities II

- The step from linear regression to logistic regression is straightforward. In the linear regression model, we have modelled the relationship between outcome and features with a linear equation:

  $$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d$$

  We can make keep an index for the instance: for any ith data instance,

  $$\hat{y}_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_d x_{i,d}$$

- For classification, we prefer probabilities between 0 and 1, so we wrap the right side of the equation into the logistic function. This forces the output to be in the range $[0, 1]$.

$$P(y_i = 1 | \boldsymbol{x}_i) = \frac{1}{1 + \exp\left(-\left(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_d x_{i,d}\right)\right)}$$

Modelling Probabilities III

- The term on the left $P(y_i = 1|\boldsymbol{x}_i)$ is read as the probability that $\boldsymbol{x}_i$ belongs to class 1.
- So, since there are two classes only (0 and 1), we can find out the probability that $\boldsymbol{x}_i$ belongs to class 0 as:

$$\begin{aligned} P(y_i = 0|\boldsymbol{x}_i) &= 1 - P(y_i = 1|\boldsymbol{x}_i) \\ &= \frac{\exp\left(-\left(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_d x_{i,d}\right)\right)}{1 + \exp\left(-\left(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_d x_{i,d}\right)\right)} \end{aligned}$$

- In a vectorised form, we can write the above as:

$$P(y_i = 0|\boldsymbol{x}_i) = \frac{1}{1 + \exp(-\boldsymbol{\beta} \cdot \boldsymbol{x}_i)}$$

where, $\boldsymbol{x}_i = [x_{i,0}, x_{i,1}, \ldots, x_{i,d}]^\top$ with $x_{i,0} = +1, \forall i$; and $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_d]^\top$.

Modelling Probabilities IV

- Now, if we divide these two probabilities $P(y_i = 1|\boldsymbol{x}_i)$ and $P(y_i = 0|\boldsymbol{x}_i)$, we get:

$$\frac{P(y_i = 1|\boldsymbol{x}_i)}{P(y_i = 0|\boldsymbol{x}_i)} = \exp\left(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_d x_{i,d}\right)$$

  The lefh hand side is called the odds or odds ratio.

- Taking the log on both sides (to cancel the exponential on the right hand side), we get:

$$\log \frac{P(y_i = 1|\boldsymbol{x}_i)}{P(y_i = 0|\boldsymbol{x}_i)} = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_d x_{i,d}$$

- So, on the right, we see that it is just the linear regression equation. And, the right hand side is called the log odds or logit.

Modelling Probabilities V

- So, in a way, what we are seeing is that: the logistic function converts the log odds into probabilities. Therefore, the name is 'logistic regression' (in a way, the linear regression is still at the core of it, but wrapped beautifully by a logistic function).
- As you notice, the discussion is restricted to binary classification problem only.

How to extend it to problems with multiple classes?

Modelling Probabilities VI

Multiclass classification: One-vs-all (one-vs-rest) Training

- Multiclass classification is implemented by training multiple logistic regression classifiers, one for each of the (say K) classes in the training dataset.
- For example, let's say there are 3 classes. We need to train 3 different logistic regression classifiers.
  - When training the classifier for class 1, we will treat input data with class 1 labels as positive samples ($y = 1$) and all other classes as negative samples ($y = 0$).
  - Similarly, when training the classifier for class 2, we will treat input data with class 2 labels as positive samples ($y = 1$) and all other classes as negative samples ($y = 0$).
  - Continue for all the classes.

Modelling Probabilities VII

Multiclass classification: One-vs-all (one-vs-rest) prediction

- Pass the data instance ($\boldsymbol{x}$) to all the trained logistic regression classifiers.
- Obtain the probabilities from the classifiers.
- The one-vs-all prediction function picks the class for which the corresponding logistic regression classifier outputs the highest probability
- Return the class label $\in \{1, 2, \ldots, K\}$ as the prediction for the input data instance.

In Practice

Practice the laboratory assignment available in the notebooks shared in the GitHub lab directory of this course. There are two problems:

- dummy dataset
- Wisconsin Breast Cancer dataset

Lab link: L07 (Logistic Regression)