6COM2000 Advanced Artificial Intelligence

Logistic Regression for Sentiment Analysis

Olga Tveretina

Overview

Linear Regression vs Logistic Regression

Linear vs Logistic Regression

Similarities:

1. Linear Regression (Least Squares is one of the models) and Logistic Regression both are supervised Machine Learning approaches.
2. Both use linear equations for predictions.

Differences:

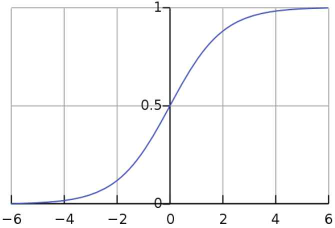
1. Logistic Regression is a classification algorithm, used to classify elements of a set into two groups (binary classification). Linear Regression is used to handle regression problems.
2. Linear Regression provides a continuous output whereas Logistic Regression provides discreet output.
3. Linear Regression finds the best-fitted line. Logistic Regression is fitting the line values to the sigmoid curve.

Foundations: Sigmoid and Tahn Functions

Useful Functions: Sigmoid

1. A sigmoid function is a mathematical function having a characteristic "S"-shaped curve or sigmoid curve.
2. A common example of a sigmoid function defined as

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



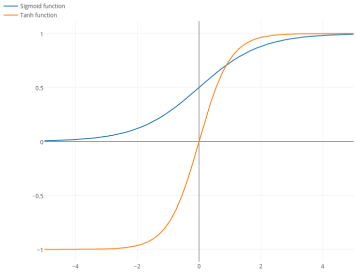
Source: Wikipedia

This function maps inputs to a range of 0 to 1. It is used, for example, in binary classification.

Useful Functions: Tahn

Tanh (Hyperbolic Tangent): Maps inputs to a range of -1 to 1, similar to sigmoid but centered at 0:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



Source: Stackexchange

You can use tanh instead of a sigmoid function. If you want to find output between 0 to 1 then we use sigmoid function. If you want to find output between -1 to 1 then we use tanh function.

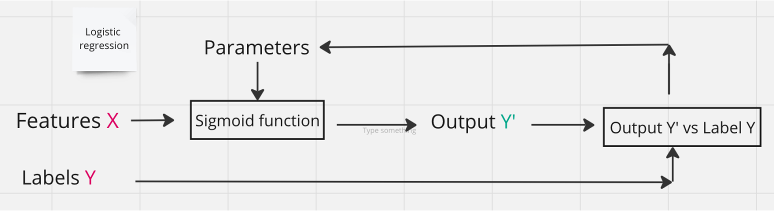
Introduction to Logistic Regression

What is logistic regression?

1. Logistic regression is a very important tool used in many applications in NLP.
2. Logistic regression algorithms are particularly useful because they are easy to train and provide you with a good baseline result.
3. Logistic regression estimates the probability of an event occurring, such as voted or did not vote, based on a given dataset of independent variables.
4. Since the outcome is a probability, the dependent variable is bounded between 0 and 1.

Overview

1. Supervised machine learning: input features and a sets of labels.
2. A function with some parameters (for logistic regression, the sigmoid function) to map your features to output labels.
3. Check how close Y' hat is to the labels Y from your data.
4. Update the parameters and repeat the process.



Example

Example (https://en.wikipedia.org/wiki/Logistic_regression)

Let us consider the following problem:

A group of 20 students spends between 0 and 6 hours studying for an exam.

How does the number of hours spent studying affect the probability of the student passing the exam?

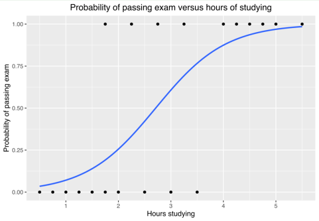
1. The values of the dependent variable, "pass" and "fail" (represented by "1" and "0"), are not cardinal numbers.
2. If the problem was that pass/fail was replaced with the grade 0–100 (cardinal numbers), then simple regression (e.g., the least squares method) analysis could be used.

Example (cont.)

Example (`https://en.wikipedia.org/wiki/Logistic_regression`)

Hours (x_k)	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass (y_k)	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

1. We wish to fit a logistic function to the data consisting of the hours studied (x_k) and the outcome of the test ($y_k = 1$ for pass, 0 for fail).
2. The data points are indexed by the subscript k, $1 \leqslant k \leqslant 20$.



Math Behind

Math Behind

1. An equation of the best fin line in linear regression:

$$y = \beta_0 + \beta_1 x$$

2. If instead of y we take the probability P then the value of P can exceed 1 or go below 0:

$$P = \beta_0 + \beta_1 x$$

3. To overcome this issue we take the "odds" of P:

$$\frac{P}{1 - P} = \beta_0 + \beta_1 x$$

Math Behind (cont.)

1. Odds are positive, that is, in the range $(0, +\infty)$. To avoid restricting the range, we take the ln of odds which has a range from $(-\infty, +\infty)$:

$$\log \frac{P}{1-P} = \beta_0 + \beta_1 x$$

2. We obtain the function of P by taking exponent on both sides:

$$e^{\ln(\frac{P}{1-P})} = e^{\beta_0 + \beta_1 x}$$

Math Behind (cont.)

And then solve for P:

$$\frac{P}{1-P} = e^{\beta_0+\beta_1 x}$$

$$P = e^{\beta_0+\beta_1 x} - Pe^{\beta_0+\beta_1 x}$$

$$P = P[e^{\beta_0+\beta_1 x}/P - e^{\beta_0+\beta_1 x}]$$

$$1 = e^{\beta_0+\beta_1 x}/P - e^{\beta_0+\beta_1 x}$$

$$1 + e^{\beta_0+\beta_1 x} = e^{\beta_0+\beta_1 x}/P$$

$$P(1 + e^{\beta_0+\beta_1 x}) = e^{\beta_0+\beta_1 x}$$

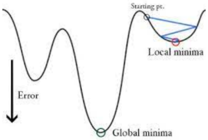
$$P = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

$$P = \frac{1}{1 + e^{-(\beta_0+\beta_1 x)}}$$

This is our logistic function, also called a sigmoid function.

Cost Function

1. A cost function can result with local minima. This is a problem because then we can miss out on the global minima and the error will increase.



Source: https://towardsdatascience.com/optimization-loss-function-under-the-hood-part-ii-d20a239cde11

2. A different cost function for logistic regression called log loss:

$$log\ \ loss = \frac{1}{n}\sum_{i=1}^{n} -(y_i \log(Y_i') + (1 - y_i)\log(1 - Y_i'))$$

3. A common evaluation metric for binary classification models. It measures the performance of a model by quantifying the difference between predicted probabilities and actual values.

Log loss (logarithmic loss or cross-entropy loss)

The log loss for the k-th point is:

$$\begin{cases} -\ln P_k \text{ if } y_k = 1 \\ -\ln(1 - P_k) \text{ if } y_k = 0 \end{cases}$$

These can be combined into a single expression:

$$-y_k \ln P_k - (1 - y_k)\ln(1 - P_k)$$

1. The sum of these, the total loss, is the overall negative log-likelihood $-\ell$, and the best fit is obtained for those choices of β_0 and β_1 for which $-\ell$ is minimized.
2. Alternatively, instead of minimizing the loss, one can maximize the positive log-likelihood:

$$\sum_{i=1}^{n} [y_k \ln P_k + (1 - y_k)\ln(1 - P_k)]$$

Parameter estimation

Optimum Coefficients

1. Since ℓ is nonlinear in β_0 and β_1, determining their optimum values will require numerical methods.
2. Note that one method of maximizing ℓ is to require the derivatives of ℓ with respect to β_0 and β_1 to be zero:

$$0 = \frac{\partial \ell}{\partial \beta_0} = \sum_{k=1}^{K} (y_k - P_k)$$

$$0 = \frac{\partial \ell}{\partial \beta_1} = \sum_{k=1}^{K} (y_k - P_k) x_k$$

3. Solve the above two equations for β_0 and β_1.

Parameter Estimation: Example

Example

The values of β_0 and β_1 maximising ℓ and L using the above example are:

$\beta_0 \approx -4.1$

$\beta_1 \approx 1.5$

Predictions

1. The β_0 and β_1 coefficients may be entered into the logistic regression equation to estimate the probability of passing the exam.
2. For example, for a student who studies 2 hours, entering the value $x = 2$ into the equation gives the estimated probability of passing the exam of 0.25:

$$t = \beta_0 + \beta_1 x \approx -4.1 + 2 \times 1.5 = -1.1$$

Then the probability of passing exam is:

$$P = \frac{1}{1 + e^{-t}} \approx 0.25$$

3. For the student who studied 4 hours:

$$t = \beta_0 + \beta_1 x \approx -4.1 + 4 \times 1.5 = 1.9$$

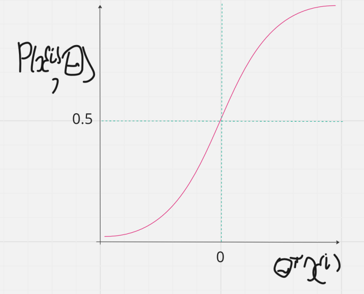
$$P = \frac{1}{1 + e^{-t}} \approx 0.87$$

Logistic Regression for Sentiment Analysis

Overview

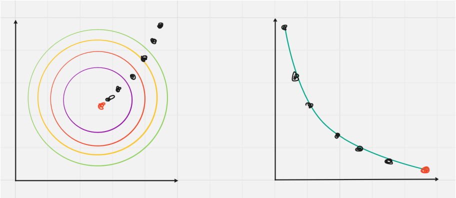
Given a tweet, you can transform it into a vector $x^{(i)}$ and run it through your sigmoid function to get a prediction:

$$P(x^{(i)}, \Theta) = \frac{1}{1 + e^{-\Theta^T x^{(i)}}}$$



Logistic Regression: Training

1. You initialize your parameter Θ, that you can use in your sigmoid
2. Compute the gradient that you will use to update Θ
3. Calculate the cost
4. Keep doing so until good enough



Testing

1. To test your model, you run a subset of your data on your model to get predictions.
2. The predictions are the outputs of the sigmoid function.
3. If the output prediction$=P(x^{(i)}, \Theta) \geq 0.5$, assign it to a positive class. Otherwise, assign it to a negative class.
4. Accuracy is

$$\sum_{i=1}^{n} \frac{prediction^{j} == y_{val}^{j}}{n}$$

Evaluating Performance

Confusion Matrix

	Negative	Positive
Negative	TN	FP
Positive	FN	TP

1. TN: True Negative which shows the number of negative examples classified accurately.
2. TP: True Positive which indicates the number of positive examples classified accurately.
3. FP: False Positive value, i.e., the number of actual negative examples classified as positive; and
4. FN: a False Negative value which is the number of actual positive examples classified as negative.