Project 3 Wisconsin Cancer data and Logistic regression and Feed Forward Neural Networks

Annika Eriksen

December 16, 2020

Abstract

The wisconsin cancer data is a staple of machine learning and should therefore be a good point to analyse different methods. I analyze the data using 2 different methods, firstly through logistic regression and later on using a Feed Forward Neural Network

1 introduction

The Wisconsin Cancer dataset is a widely used set for classification problems, and offers a good starting point to compare implementations of machine learning methods. I have put the set through 2 different methods, Logistic regression and a Feed Forward Neural Network.

First, the Logistic regression takes in the data and feeds it into the sklearn logistic regression method. This is done without- and then with scaling the data to see prediction accuracy depending on these. Then I take only those features with the greatest correlation to a malignant tumour and train the set on those only, to see the predictive ability of the methods based on these.

While we see an expected increase in accuracy after normalising the features, there is a rather unexpected decrease in accuracy when only including the more important features.

Then, Implementing a Feed Forward Neural Network (ffnn), We construct a network with 1 hidden layer and a binary input, either malignant or benign. The initial model is then iterated a number of times, or until the predictive accuracy of the model on the test set has passed it's peak.

As one expects, the neural net quickly goes toward an above 90% accuracy for both training and test set until eventually overfitting sets in and the test set accuracy starts to fall off.

2 methods

The base problem to be solved with the cancer data set is a binary benign vs. malignant tumor, 0 or 1 encoded respectively,

$$y = \begin{bmatrix} 0 & \text{benign} \\ 1 & \text{malignant} \end{bmatrix}. \tag{1}$$

We have 2 *classes*, which form the entirety of the possible outcomes within the set. Like with regression I want a model to give an expected value for a given set of samples for the features. A way to model would be to assume some linear functional dependence, along the lines of

$$f(y_i|x_i) = \beta_0 + \beta_1 x_1.$$

Although this could contain values outside of 0 and 1. Taking the mean solves this and forces $f(x|y) \in [0, 1]$. This then can be seen as the probability for finding a given value for y_i with a given x_i. This S-shaped function would give us the *sigmoid* function. The soft predictor gives a probabilistic prediction of the class of outcome, given the input feature. In my case, this is either of the options in y in eq 1. The Sigmoid, or *logit* function governs this probability prediction for a given event thus,

$$p(t) = \frac{1}{1 + \exp(-t)} = \frac{\exp(t)}{1 + \exp(t)}. \tag{2}$$

Noting, $1 - p(t) = p(-t)$.

Since we have 2 possible outcomes, the probabilites are then given by

$$p(y = 1|x_i, \hat{\beta}) = \frac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)} \tag{3}$$

$$p(y = 0|x_i, \hat{\beta}) = 1 - p(y = 1|x_i, \hat{\beta}). \tag{4}$$

Denoting the biases with **fi** which I want to extract from the dataset.

Using the Maximum Likelihood Estimation principle The total probability for all possible outcomes of a dataset $\mathcal{D} = (y, \mathbf{x_i})$, with the binary in eq. 1 for each y_i. The probabilities can be approximated by a product of the individual probabilities for an outcome of y_i,

$$\mathbf{P}(\mathcal{D}|\hat{\beta}) = \prod_{i=1}^{n} \left[p(y_i = 1|\hat{\beta})\right]^{y_i} \left[1 - p(y_i = 1|\hat{\beta})\right]^{1-y_i} \tag{5}$$

And from this we can find the log-likelihood and our *loss* function.

$$\mathcal{C}(\hat{\beta}) = \sum_{i=1}^{n} \Big(y_i \log\left[p(y_i = 1|x_i\hat{\beta})\right] + (1 - y_i) \log\left[1 - p(y_i = 1|x_i\hat{\beta})\right] \Big). \tag{6}$$

Inserting for the probabilities with eq. 3 and sorting the logarithms, we can rewrite the loss function. The cost/error function is the negative log-likelihood function, so it would be the negative of this rewrite. The end result then, is

$$\mathcal{C}(\hat{\beta}) = -\sum_{i=1}^{n} (y_i(\beta_0 + \beta_1 x_i) - \log(1 + \exp(\beta_0 + \beta_1 x_i))). \tag{7}$$

This is also known as the cross-entropy. This can be further supplemented with regularization parameters much like with Lasso and Ridge regression.

The next step then is to minimize the cost function. Differentiating wrt. β_0 and β_1 and compacting, gives us

$$\frac{\partial \mathcal{C}(\hat{\beta})}{\partial \hat{\beta}} = -\hat{\mathbf{X}}^T(\hat{y} - \hat{p}). \tag{8}$$

Where $\hat{y}$ is a vector with the values y_i, $\hat{\mathbf{X}}$ is an $n \times p$ matrix containing the x_i values. The vector $\hat{p}$ contains the fitted probabilities $p(y_i|x_i, \hat{\beta})$. For a step further and introducing another matrix, $\hat{\mathbf{W}}$ with elements $p(y_i|x_i, \hat{\beta})(1 - p(y_i|x_i, \hat{\beta})$, we can arrive at a more compact version of the 2. derivative.

$$\frac{\partial^2 \mathcal{C}(\hat{\beta})}{\partial \hat{\beta} \partial \hat{\beta}^T} = \hat{\mathbf{X}}^T \hat{\mathbf{W}} \hat{\mathbf{X}} \tag{9}$$

3 results

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

4 conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut,

ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.