Solving Classification Problems with Logistic Regression

Mathew Roberts

January 2021

1 Introduction to Classification Problems

Classification problems are a subset of Machine Learning whereby algorithms are created to prescribe discrete 'class labels' to input examples within the domain. For example, a simple case would be identifying the biological sex of a person (Male or Female) taking into account their height and weight.

In the following examples, classification problems such as the one above will be discussed and analysed. Examples include, choosing the correct wine name given specific attributes about that wine and the well-known MNIST hand-written digit identification problem.

It is important to note that two different aspects of classification will be discussed, binary and non-binary classification. The former concerns a problem in which the class labels are restricted to only two values; an example of this would be the height-weight classification problem previously mentioned, as the class labels can only be an option of 0 (Male) or 1 (Female). However, the wine classification problem and the MNIST problem both have more than two possible class labels; hence, the approaches are slightly different.

In this report, a variety of different classification techniques will be employed, however, they all stem from logistic regression. Binomial logistic regression will be used for the height-weight example, whereas multinomial logistic regression will be used for the wine data-set and the MNIST problem. To guarantee the best model performance for our classification problems, various regularisation techniques such as Ridge and LASSO regularisation will be used.

2 Brief Theoretical Background: Logistic Regression

2.1 Binary Logistic Regression

For continuous data-sets it is common to use a cost function that uses the sum of the squared residuals of the data-set. An example of this is the Mean-Square Error (MSE), shown in Eq.1,

$$MSE(w) := \frac{1}{2s}\sum_{i=1}^{s}(f(w,x_i)-y_i)^2. \tag{1}$$

If a binary classification case is considered, one can label the classes 0 and 1 respectively. However, Eq.1 will punish large estimates of y_i (i.e $y << 0$ or $y >> 1$), despite the fact that large estimates mean that the model is overly confident of the class prescription. The MSE cost function is not related to the objective of minimising the number of mis-classified samples, and therefore it is not suitable for classification problems. It is more reasonable to transform the predicted y-values (given by the model function $f(w,x_i)$) into a probability, rather than having $f(w,x_i) \in (-\infty,\infty)$. Hence, we introduce the logistic function,

$$\sigma(z) := \frac{1}{1+\exp(-z)}, \tag{2}$$

where, $\sigma(f(w,x_i)) \in [0,1]$. This has now mapped the model function to a probability, whereby one can impose a condition on the probability outcomes to deduce whether the data point is likely to belong to class 0 or 1.

The optimal weights, w, are found via a gradient descent algorithm derived from the cost function shown in Eq.3,

$$L(w) := \sum_{i=1}^{s} [\log(1 + \exp(\langle \phi(x_i), w \rangle)) - y_i \langle \phi(x_i), w \rangle]. \tag{3}$$

2.2 Multinomial Logistic Regression

Multinomial logistic regression is used when there are more than two distinct class labels. In order to construct this model, we require the softmax function,

$$\sigma(z_1, z_2, z_3, ...z_K)_k := \frac{\exp(z_k)}{\sum_{j=1}^{K} \exp(z_j)}, \forall \in \{1, ..., K\}. \tag{4}$$

The softmax function, when acting upon an ideal model function, will assign the highest probability to the label that corresponds to the correct class for each data point. The optimal weights w follow similarly to the method in section 2.1, whereby a gradient descent algorithm is formed from the cost function given by Eq.5

$$L(W^k) := \sum_{i=1}^{s} \log(\sum_{i=1}^{K} \exp(\langle \phi(x_i), w_j \rangle)) - \sum_{i=1}^{s} \sum_{j=1}^{K} 1_{y_i=j} \langle \phi(x_i), w_j \rangle \tag{5}$$

2.2.1 With Ridge/LASSO regularisation

In Ridge regularisation, weight elements with minor contributions have their coefficients altered to be close to zero; whereas in LASSO regression, the weight elements with less significance are forced to be exactly zero - only the most relevant variables are kept in the final model. The regularisation strength of these parameters is governed by hyper-parameter α.

When using regularisation techniques, the above cost function (Eq.5) is amended via an addition of a single term; this is $\frac{\alpha}{2}\|w\|_2^2$ for the Ridge regression case, and $\alpha\|w\|_1$ for the LASSO regression case. The latter is non-differentiable, thus it must be solved via proximal gradient descent.

3 Results of the Classification Problems

3.1 Height-Weight-Sex Binary Classification

The height-weight-sex classification problem required the use of binary (or binomial) logistic regression, as described in section 2.1. The data-set comprised of 10,000 sample pairs of height and weight, with the corresponding sex's assigned in a separate array. The data set comprising of the sample pairs of height and weight were standardised and then converted into a polynomial basis for further processing. A gradient descent approach, derived from Eq.3, was used to find the optimal weights for the model. Regularisation methods were not used in this simple example as the task was to obtain the best classification accuracy on the training set.

With a polynomial basis matrix of degree = 1 (X), and optimal weights (w), a prediction model $f(x, w) = Xw$ was formed. The gradient descent was carried out over 7000 iterations, by which time a $1\text{x}10^{-10}$ difference between adjacent iterations had been achieved. An accuracy of 91.94% was subsequently obtained on the training set.

3.2 Wine Classification

The wine classification problem required the use of multinomial logistic regression, as the number of class labels had increased from 2 to 3 rendering the binomial approach invalid. The wine data-set consisted of 178 samples, with each sample containing 14 attributes describing the wine, as well as the specific class labels of each sample. The input data for the 178 samples were standardised and converted to a polynomial basis

for further processing. Similarly, to the height-weight-sex classification problem in section 3.1, a gradient descent approach was required to find the optimal weights for the model; however, both Ridge and LASSO techniques were used in addition to a non-regularised logistic regression method.

For the non-regularised method, the number of iterations chosen for gradient descent was set at 100,000. After this many iterations, a classification accuracy of 100% was obtained for the training set. This is indicative of over-fitting and the model function $f(x, w) = Xw$ is likely to have a large variance in validation error when used for other test sets.

The next model tested was the Ridge regularisation model. Here, a regularisation strength α was chosen to be 15. After 100 iterations the cost function had converged to only a $1\text{x}10^{-4}$ degree of accuracy between previous iterations, therefore the iterations were halted. The ridge classification accuracy obtained was 99.4%, slightly less than the non-regularised method. However, it is important to note that the data-set was rather trivial and consisted of a very small number of samples therefore a high degree of accuracy was to be expected.

Finally, a LASSO regression method was used to obtain optimal weights for model function $f(x, w)$. An alternative form of gradient descent, proximal gradient descent, was used as the additional term in LASSO is non-differentiable. After 10000 iterations had been performed, the cost function had only converged to a $1\text{x}10^{-2}$ difference between adjacent iterations; nonetheless, a classification accuracy of 100% was still achieved. Once again this classification accuracy is remarkably high, and is likely to be over-fitting the data, perhaps this is due to the regularisation parameter ($\alpha = 0.5$) being very small so it doesn't penalise the model function.

3.3 MNIST Hand-written Digits

The MNIST database is a large database of hand-written digits which are most commonly used for training image processing models. It contains 60,000 training images, each containing 28x28 (784) pixels. The efforts detailed below attempt to obtain the highest classification accuracy from a model $f(x, w) = Xw$ using different regression techniques to tune the weight parameters. The above regression techniques will be revisited in an attempt to achieve this goal.

The 60,000 sample MNIST data-set was carefully standardised, making special adjustments for the columns where the standard deviation was zero, then the data-set was partitioned into two main sets, the validation set and the testing/training set. This was done to ensure that a substantial amount of data had not been used for training so that over-fitting was avoided. The partition was balanced via an 80—20 split, where 80% was used for training/testing, and 20% was withheld from all training models.

The training/testing set was further split, where 70% was used for training and 30% was used for testing. Thus from a total of 60,000 samples, 12,000 were withheld as a validation set, leaving 48,000 samples for testing/training. The further splitting, meant that the training sets consisted of 33,600 training samples and 14,400 test samples.

3.3.1 Non-regularised Multinomial Logistic Regression (NR MLR)

The first regression model was a simple multinomial logistic regression model. Taking into account the computational limitations as well as the size of the data-set, it was decided that a simple test-train split as mentioned in section 3.3 was sufficient for this computation. If cross validation were to be considered , the number of iterations would have to be lowered for each K training cycle, therefore hindering the ability to find the global minimum of the multinomial cost function.

The model was trained using 33,600 randomised data samples from the 48,000 test-train samples removed from the overall set. Randomisation of data samples removes any selection bias that may seek to invalidate the models created. Unfortunately, computational limitations meant that only a maximum of 500 iterations were used, however this still led to a convergence of < 100 between adjacent objective values; considering the objective values were on a magnitude of 10^5 this was acceptable. Testing the accuracy on the test labels (which had not been used for training) yielded an impressive accuracy of 90.65%; this suggested that the

multinomial logistic regression without regularisation does not have an issue of over-fitting. This hypothesis was confirmed by a classification accuracy of 90.95% on the validation set.

Label	0	1	2	3	4	5	6	7	8	9
Classification Accuracy (%)	97.45	96.55	87.90	87.23	93.26	82.70	94.88	93.33	84.86	89.07

Table 1: This table shows the classification accuracy's for different digits. The model used to obtain these values had a polynomial basis of degree = 1. These values are from the validation data-set

Table 1 shows the classification accuracy per label. We can see that the numbers 0,1,4,6 and 7 were classified the best, and 5,3 and 8 were the worst with the number 5 being the worst accuracy of them all. Perhaps this is due to the many drastically different ways that people have written these numbers, whereas 0,1,4,6 and 7 are prone to less variance.

3.3.2 Changing the Degree of the Polynomial Basis (NR MLR)

This section describes the effect of changing hyper-parameter d of the polynomial basis $\Phi(X)$. The previous section, 3.3.1, had a predictive model function of the form $f(X, w) = \Phi(X)w$, where $\Phi(X)$ was a polynomial of degree one; in this section, polynomial basis' of degree 2, 4 and 6 will be tested.

Polynomial Degree (d)	Classification Accuracy
2	64.63%
4	49.70%
6	35.34%

Table 2: This table shows the classification accuracy's for different degrees of the polynomial basis (Test Set).

Table 2 implies an overall decrease in classification accuracy with increasing polynomial degree. Its important to note that only a small sample of degrees were tested, therefore this trend may not hold for all possible degrees. A polynomial basis of degree 2 results in the highest classification accuracy of the test set. However, the accuracy of 64.63% is much worse than the performance of a polynomial matrix of degree = 1(90.95%). The accuracy achieved on the validation set was slightly higher at 66.33% - however, this is still far from optimal.

Label	0	1	2	3	4	5	6	7	8	9
Classification Accuracy (%)	93.21	90.63	62.79	70.49	31.79	42.18	86.58	88.70	28.37	37.20

Table 3: This table shows the classification accuracy's for different digits. The model used to obtain these values had a polynomial basis of degree = 2. These values are from the validation data-set

From table 3, we can see that changing the degree of the polynomial basis had a significant impact across all labels. Digits 8,9,4 and 5 all had classification performances less than 50%, however what is surprising is that digits 0,1,6 and 7 were all classified with a high degree of accuracy. There is significant overlap with the model from section 3.3.1 in terms of which digits are classified most correctly. Only digit 4 is the outlier, as it was accurately classified in section 3.3.1, but badly mis-classified when using a polynomial basis of degree = 2.

3.3.3 Ridge Multinomial Logistic Regression

Without regularisation, there is always a concern that your model function can be over-fit to your data-set. There is no guarantee that an alternative test-set will not show a large classification error between the

recovered class labels and the true class labels. Therefore, introducing a penalisation term to decrease the variance in validation error at a small expense in bias will lead to the best generalised model.

Three different regularisation strength parameters were used; these were chosen to be (1,10,100). Due to computational limitations, a broad range of regularisation parameters were used to increase the scope of the training method; this allowed for the correct magnitude of regularisation strength to be chosen, but further detail beyond this was not attained. The size of the regularisation strength parameter only minimally changed the classification accuracy obtained at the end. For each regularisation strength parameter, the method was trained over 500 iterations in order to be consistent with the computations from 3.3.1.

Regularisation Strength (α)	Classification Accuracy
1	90.65%
10	90.65%
100	90.60%

Table 4: This table shows the classification accuracy's for different regularisation strengths.

Table 4 shows a summary of the classification accuracy's (of the test set) obtained with each α. The optimal weight matrix used for the validation data-set, was the matrix constructed with regularisation strength 1.

This yielded a classification accuracy of 90.95% on the validation data-set - slightly higher than the classification results on the test data-sets. We can clearly see that the classification accuracy's obtained through non-regularised, and regularised multinomial logistic regression are very similar, this suggests that the bias introduced by the penalisation term is fairly small.

Label	0	1	2	3	4	5	6	7	8	9
Classification Accuracy (%)	97.45	96.55	87.90	87.23	93.26	82.70	94.88	93.33	84.86	89.07

Table 5: This table shows the classification accuracy's for different digits. The model used to obtain these values was a Ridge regularised multinomial logistic regression model. These values are from the validation data-set

Table 5 shows the classification accuracy for each hand-written digit. As expected, the values are exactly the same as the values in table 1. This shows that regularisation has not had much of an effect on the generalisation accuracy for an MNIST data-set.

3.3.4 LASSO Multinomial Logistic Regression

LASSO regularisation is a stronger shrinkage method than Ridge Regression, meaning that less impactful coefficients are shrunk to exactly 0. This leads to a smaller variance in the model, which helps to improve the model's accuracy on unseen data. The regularisation strength parameters for LASSO were chosen to be (1,10,100). The proximal gradient descent method was used to minimise the cost function (Eq.5) to obtain optimal weight matrix w, and a polynomial data-matrix of degree = 1 was used to form the model.

Table 6 shows the LASSO classification accuracy's for different regularisation parameters. The regularisation strength had little difference on the classification accuracy - only a 0.03% difference was observed despite the regularisation parameter increasing 10-fold.

Regularisation Strength (α)	Classification Accuracy
1	90.60%
10	90.60%
100	90.57%

Table 6: This table shows the classification accuracy's for different regularisation strengths.

The optimal weight matrix corresponded to the model with regularisation strength = 1. When the model was tested on the validation data-set, this resulted in a classification accuracy of 90.84%. This is 0.11% less than the classification accuracy of Ridge regression and multinomial (degree = 1).

Label	0	1	2	3	4	5	6	7	8	9
Classification Accuracy (%)	97.54	96.55	87.57	86.89	93.34	82.88	94.96	93.33	84.07	88.98

Table 7: This table shows the classification accuracy's for different digits. A lasso model with regularisation strength = 10 was used to create these values. These values are calculated from the validation data-set

From table 7, we can see that there is a net decrease in classification accuracy across all labels. This is consistent with the overall classification (90.84%) being less than Ridge and multinomial(degree = 1). The trend of which digits the model has most correctly classified has also remained the same.

4 Conclusion

Logistic regression techniques were employed to provide the best predictor model for the MNIST data-set. Regularisation techniques such as LASSO and Ridge regression were tested, as well as hyper-parameter tuning strategies for the polynomial basis degree d. Using the classification accuracy from the multinomial (degree = 1) as reference, it was found that Ridge regression had no effect on the classification accuracy unless large regularisation strength parameters were used, LASSO regression slightly decreased the accuracy by 0.11% and increasing the polynomial basis degree drastically reduced the classification accuracy by almost 30%. Thus, the best model which has been found in this report is the model - non-regularised multinomial (degree = 1).

All tested models had similar difficulties in classifying particular digits. These digits were (2,3,5,8) which all had classification accuracy's of under 89%, whereas digits (0,1,4,6 and 7) were classified accurately ($>$ 93%) for all models. This could be due to the fact that digits (2,3,5,8) have more variations in how they are written, therefore the learning models had trouble identifying those digits consistently.