Gaussian Naive Bayes and Logistic Regression

Joe Shymanski

March 30, 2022

1 Introduction

Gaussian Naive Bayes (GNB) and Logistic Regression (LR) are very similar models for machine learning classification problems. The key difference is that GNB is a generative model while LR is discriminative. Thus, the two models may not perform identically on the same data set. These differences will be explored in this paper.

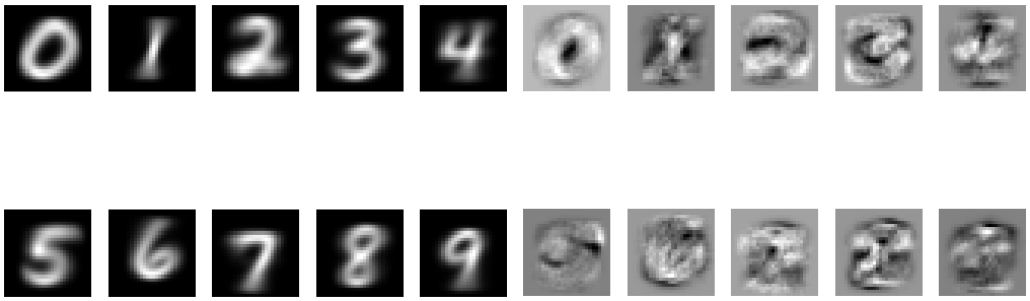
2 Data Sets

The MNIST digit-recognition data set was used to test both models [1]. Each entry consists of 784 pixel intensities from 0 to 255 for a 28×28 handwritten digit. For the sake of this paper, each pixel intensity can be expressed as a real number ranging from 0 to 255, even though the data set only consists of integer values.

3 Models

The GNB model contains three important sets of data. The first is the proportion of each label in the training data. The second is a table of means for each feature given each label. The third is nearly identical to the second, but containing standard deviations instead of means. These are all generated from the training data made up of n examples, each with m features in the form $X = \langle X_1, X_2, ..., X_m \rangle$ and a label Y.

In order to predict the data, the Normal PDF is computed for each feature given each label across all examples. The natural logarithm is applied to



(a) GNB (b) LR

Figure 1: Visualizations for both models.

each of these probabilities, which are then summed across all features within a label. The natural log of the label's probability is then added to this sum. The label with the maximal sum is picked as the predicted label for the test example.

For LR, instead of generating the intermediate probabilities $P(X|Y)$ and $P(Y)$ to compute the final predictions, it seeks to compute $P(Y|X)$ directly. Thus, its model simply contains a list of weights $\mathbf{w} = \langle w_0, w_1, ..., w_m \rangle$, one for each feature. To find the optimal weights, gradient descent is utilized with the update rule by the following equations:

$$w_i = w_i - \eta\lambda w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, \mathbf{w})) \tag{1}$$

$$\hat{P}(Y^l = 1|X^l, \mathbf{w}) = \frac{\exp(w_0 + \sum_i w_i X_i^l)}{1 + \exp(w_0 + \sum_i w_i X_i^l)} \tag{2}$$

where η is the learning rate and λ is the regularization parameter. The weights are updated for a set number of epochs or if the update is less than a given tolerance level.

Since the problem contains more than two classes, LR will need to be modified to a one-vs.-all method of classification. This means that a separate $\mathbf{w}$ will be calculated for every class y_k, where $Y^l = 1$ if $Y^l = y_k$ and $Y^l = 0$ otherwise. To make a prediction, the model computes $\hat{P}(Y^l = 1|X^l, \mathbf{w})$ for each class's list of weights and returns the class with the largest probability.

Figure 1 gives an example of a visual representation for both models. In Figure 1a, the mean values for each pixel clearly show what an average picture of each digit should looks like. In Figure 1b, the weights do not paint

Model	Train Accuracy (%)	Test Accuracy (%)	Time (s)
GNB	64.89	64.72	9.16
LR (reg.)	77.48	76.98	49.46
LR (no reg.)	81.83	81.44	51.31

Table 1: Model performance results.

such a clear picture. For digits like 0, 2, or 3, you can see that the darker pixels (negative weights) form edges around many lighter pixels (positive weights) to create a thicker representation of each digit.

4 Experimental Results

Table 1 reports the results for each model using 5-fold cross-validation. For LR, $\eta = .2$ and the number of epochs was set to 200. For LR with regularization, $\lambda = .02$. All 42,000 examples were used with each model, meaning 33,600 examples were seen in each training set. In order to reproduce the results, the seed for the NumPy random number generator was set to 517. Each accuracy and time represents the average of all five runs from cross-validation.

5 Discussion

As is clear in Table 1, GNB took a fifth of the time that LR needed to train a model and make predictions on the same number of examples. This difference in classification time is likely due to the sheer difference in complexity in training. GNB simply takes averages and standard deviations over the whole data set, both of which are incredibly fast and optimized calculations. LR, however, must do some form of iterative gradient descent to reach a potentially suboptimal solution for the model weights. Most of this can be sped up using vectorized functions instead of explicit loops, but it will never be as efficient as GNB training.

Figure 1 provides a summary, albeit imperfect, of the resulting model complexities. GNB simply creates an average shape for each digit, but LR detects a bubble of acceptability with some pockets of negative weights sur-

rounding the edges. The latter appears to be a bit more complex and potentially more robust model than the former. LR also does not require the independence assumption that GNB needs to perform well, and pixels on a screen are highly correlated with their neighbors.

Table 1 also shows that LR is clearly more accurate than GNB, while neither model seemed to be overfitted. I had anticipated a drop in model efficacy for LR given its one-vs.-all modification for this non-binary problem, but this was not the case. Its model, which contains less parameters than GNB but a higher training time, proved to be far more powerful.

Finally, the results show that the LR model with no regularization performed slightly better than the model with regularization. The reason for this may be due to the fact that not all features are actually close to zero in value. The regularization parameter in this model is used to reflect that prior assumption about the data.

6 Conclusion

The GNB and LR models performed fairly well for classifying handwritten digits, especially considering that a null model would only classify about 10% of the data set correctly. Using the same number of examples, LR takes about five times longer than GNB to train and predict, but it's significantly more accurate, especially with optimal parameters. For this problem, LR without regularization seems preferable to LR with regularization, as the underlying prior assumption does not seem to hold.

References

[1] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.