Nicholas Louis Brown

Professor Wallace

IST 707

October 17, 2018

Digit Recognition Using Multinomial Naïve Bayes Algorithm

**Executive Summary:**

Multinomial Naïve Bayes is a versatile algorithm with many applications. In this report I used MNB to train a digit recognition model. Data for the model was created by generating a 28x28 grid and recording the pixel values for each individual pixel along with a label column containing the digit answer. There are currently multiple OCR solutions so ours is not particularly novel however this exercise is a great example of the capabilities of MNB for image recognition/prediction. Much can be learned from the process of converting images into trainable data for the algorithms. MNB ended up outperforming decision tree by a sizable margin again showing the great power of Bayesian statistics.

**Introduction:**

Data for training the model was provided by Kaggle for a handwritten digit recognition competition. The data contains 785 attributes (label and pixel0-783) and a total number of 1400 observations. Distribution of the data is mostly even ranging from 127 (5) to 167 (1) observations. Models were evaluated using 10-fold cross validation available in Weka. In addition to the MNB model I created a J48 decision tree model to compare the limitations and benefits of each model. All model creation and analysis was done in Weka using preformatted Arff files.

**Decision Tree:**

****Using Weka’s J48 decision tree function I created a decision tree which attempted to predict the digit label. Using 10-fold cross validation I ended up with 994 or 71.00% correctly classified instances and 406 or 29% incorrect. The confusion matrix below shows where the model went wrong. Interestingly the decision tree seems to make many errors misplacing 2’s, 3’s, and 4’s for 7’s, 8’s, and 9’s. The limitations of decision tree are apparent here. J48 appears to not account for subtle differences in various handwritings leading to confusion on its part.

**Multinomial Naïve Bayes:**

As I hypothesized the MNB model was much more successful than the decision tree. After conducting 10-fold cross validation the model correctly classified 1164 instances (83.14%) and missed only 236 (16.85%), nearly half of the decision trees. Like the decision tree the confusion matrix shows problems with successful predictions in the 3, 4, and 5 range though much less severe than the decision tree. Additionally, the MNB model performs much better on average with many classifications showing no cases of mistaken identity.

**Performance Comparison:**

In terms of build times J48 comes in significantly slower taking 1.13 seconds compared to the MNB models 0.01 seconds. Additionally, cross-validation was rather slow on the J48 tree whereas the MNB model was about 20 times faster. I think the limitations of the basic statistics employed by J48 and decision trees in general are apparent here. MNB is better able to account for new unseen information where decision tree is relying on a built-out tree with no extra calculations being done.