9/26/2013

Handwriting Identification for Digits 0 - 9

Orange Team 6:

Team Lead: Phillip Domschke

Other Contributors: Marc Zimmerman, Steve Neola, Wes Ledebuhr, Jacob Frost

TABLE OF CONTENTS

EXECUTIVE SUMMARY

By analyzing almost 3,500 handwriting samples, we were able to gain valuable insights that can be used by software to aid in fraud detection, paperless customer services, and other applications. Decision trees were created and pruned to classify each digit based on three input variables that represent the principal components of 16 individual coordinates defining the figure.

Taking all of this into consideration we can conclude that the digits 9, 4, 5, and 6 are most accurately identified while 0, 1, 8, and 7 are more challenging.

ANALYSIS

Large Decision Tree

Decision trees divide a data set along input variables to identify what trends in those variables describe different outputs. When a tree reaches an endpoint, or leaf, it predicts that all inputs with that combination of inputs will have the same output. Assessing the number of incorrectly placed observations yields an accuracy measure for the ability of this tree to predict a certain digit.

With the handwriting data set, we created a large decision tree to discover basic connections for what coordinates define certain digits. This original tree contained 23 leaves classifying the 10 unique digits. The two most accurate nodes both correctly divided the training set into bins of only the desired digits. Node 14 and 15 correctly organized the digits 4 and 5 respectively at 100% accuracy. When compared to the validation data set, both of these nodes agree in the digit predicted as well, although at a lower rate. Node 14 was again 100% accurate in the validation set (it only had 1 observation), yielding 0% misclassification. Node 15 had a 7.1% misclassification rate in the validation set. Following this same logic, leaves 51, 24, and 6 are also very accurate at organizing certain digits (9, 5, and 6 respectively) at 95% accuracy or greater.

It is important to note that this decision tree organizes the data set into 23 subgroups while it is only trying to predict 10 unique digits. This is due to the fact that the region represented by the three principal components is roughly cube shaped. Every time we split the decision tree, we divide this cube into smaller and smaller prisms trying to box each digit within its own space. However, several of the digits that are more difficult to define may be spread out across this region or appear in different areas because they share characteristics with the way people write other digits. In this way, our decision tree may split a single digit into multiple clusters. This results in 23 organizational nodes instead of the expected 10.

One way to assess a decision tree is through a leaf statistic plot. This plot shows all the leaves that contain a certain digit ordered by their accuracy at predicting that digit. Ideally, we'd like to see these plots for all 10 of our digits to see how clustered each digit is across the distribution. In a perfect world, each leaf statistic plot would only contain one leaf as each digit would be perfectly identified by one subgroup. However, not only is there error, but we can only see the leaf statistic plot for the digit 9. It appears in one highly accurate node, discussed above, as well as five others that predict different values.

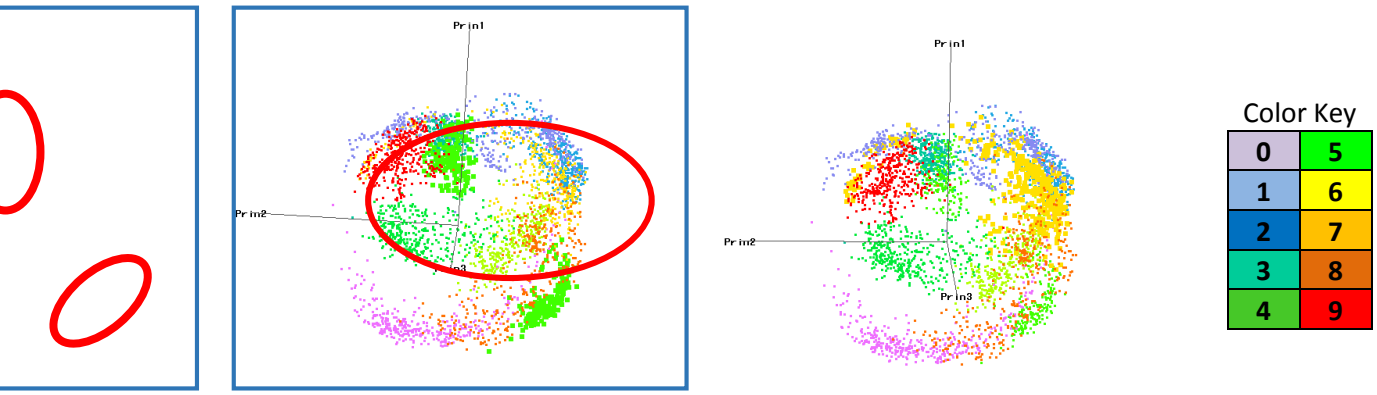
Pruned Tree

While the original tree is accurate in its predictions, it loses some of its interpretability because of the number of leaves it produces. Therefore, we built another tree that was restricted to 10 leaves to model each of the digits. The table below provides the nodes with the digits they predict along with the accuracy of each prediction in the training and validation data sets:

Predicted Digit	Node Number	Correct Predictions in Training Set	Correct Predictions in Validation Set	Misclassification
0	17	69.1%	71.9%	28.1%
1	22	69.3%	69.4%	30.6%
2	10	40.6%	42.0%	58.0%
3	48	83.9%	81.4%	18.6%
4	19	94.3%	92.3%	7.7%
5	49	87.5%	77.3%	22.7%
5	16	80.6%	82.4%	17.6%
6	3	82.1%	78.6%	21.4%
7	None	None	None	None
8	18	55.8%	61.3%	38.7%
9	37	93.1%	95.4%	4.6%

First, note that 9, 4, 6, and 5 are the digits that are best identified by this tree while 0, 1, 2, 8 are more difficult. 7 is not represented anywhere in this table. This is due to the fact that 7's are so widely spread around the principal component region that they cannot be grouped into a single cluster. As a result, the leaf that would have contained the 7's, instead creates another cluster of 5's.

3-Dimensional Plot of Input Principal Components Colored by Digit



The first plot highlights the two tight clusters of 5's (bolded in light green) that appear in the table as two unique leaves. The second plot portrays the large area covered by the 7's (bolded in orange) that cannot be organized into a concise prism cluster and demonstrate why it does not get its own leaf. These plots also depict why some digits are more easily identified (such as the 9's in a tight red grouping) and why some are more difficult (such as the 0's in a purple crescent shape).

Decision Tree Analysis

To demonstrate the use of the decision tree analysis, we predict a digit outcome for an observation with principal components PRIN1=-3, PRIN2=1, PRIN3=-1. Following down our pruned tree, this observation would fall into the node predicting the digit 0. Both the training and validation data sets agree in this prediction with a misclassification rate of 28%.

At this point, it is important to provide evidence for only using three input variables instead of the original 16 coordinates provided by the data set. A full analysis is impractical because we cannot visualize a distribution of data in 16 dimensions. Instead, we used principal components to approximate most of the variation in this set. The three principal components represent the three largest Eigen values and explain 66.3% of the variation in the original 16 coordinates. This method is much stronger than simply picking three coordinates at random yet allows 3-dimensional modeling and clustering.

Principal Component Analysis

Each of the three principal components used in this analysis is a weighted average of the 16 observed coordinates. To compute each value, we start by standardizing the input coordinates. By subtracting the mean of each coordinate and dividing by the standard deviation we create values whose mean is zero. Next, we create a linear combination of these standardized coordinates with the Eigen vectors previously calculated. This process can be repeated with each Eigen vector to reveal each of the 16 principal component values. The weights for this linear combination are shown below:

Variable	Prin1 (Eigen Vectors)
X1	-0.030097
X2	-0.151706
X3	0.250987
X4	0.318718
X5	0.227477
X6	0.418475
X7	0.116437
X8	0.377034
X9	0.037485
X10	0.118399
X11	-0.051928
X12	-0.295825
X13	-0.156495
X14	-0.418838
X15	-0.011823
X16	-0.35105

Model Comparison

We also wanted to compare the principle components model outlined above against other possible decision tree models for handwriting recognition. To do so, we created 2 additional models: one that contained principle components 1 thru 5, and another that contained the variables X1 thru X16. Each set of variables was used to create a maximal tree as well as a pruned, 10-leafed tree. The comparisons of the models based on their respective misclassification rate in the validation data set are outlined in the following tables:

Model Comparison: Maximal Tree	
Model	***Misclassification Rate***
X1-X16	0.0996
PrinComp1-5	0.1338
PrinComp1-3	0.1983
Model Comparison: Pruned Tree	
Model	***Misclassification Rate***
PrinComp1-5	0.3093
PrinComp1-3	0.3149
X1-X16	0.3283

Based on the criterion of misclassification rate, the model including the X variables outperforms the principle components models when allowed to produce a maximal tree. However, when the number of leaves is restricted, the principle components models outperform the X variable model.

CONCLUSION

When compared to the other decision trees in this analysis, the large principal component tree had the largest ROC index (98.5) followed closely by the pruned version (98.2). This demonstrates that by pruning down to 10 leaves we do not lose a lot of accuracy while gaining a lot more interpretability. For comparison sake, we also make two decision trees using the 16 coordinates as inputs instead of the principal components. However, these models had smaller ROC indexes of 96.8 and 94.4 respectively indicating that while good, our previous models are better.

Using several modified decision trees for predicting handwritten digits, a model is constructed to predict nine out of the ten digits. The following conclusion is reached – The digits 4, 9, 5, and 6 are more accurately predicted than 0, 1, 7, and 8.

Our findings suggest that principal component analysis is a viable method to help model hand writing samples. These simpler models will be more easily integrated into applications that can help identify potential fraud and analyze paperless customer services.