Case Study:
Letter Recognition Data

Kaylee Hodgson and Jacob Merrell

Brigham Young University

Abstract. Digitizing hand-written historical records has become an important way to maintain, and make more widely available, historical information. The digitization process has generally been performed by scanning the documents or having people read and manually type them in, where both of these methods have high risk of error, and the second method is labor-intensive. In this study we propose a prediction model to help computers better recognize hand-written content. We build a random forest model with 19,999 data points that indicate the "person verified" letter, as well as characteristics of that letter. The model outputs predictions for letters in documents when fed each letter's characteristics. This model is a useful tool that we hope will reduce error in digitizing historical documents, and will require less labor-intensive efforts from people who have previously had to manually type in the content.

1 Introduction and Background

In our now digital world, most important information is stored on computers. However, that option for recording has, of course, not always been available. Many of the important historical records we have are hand-written. In an effort to record and preserve these hand-written historical records, people have worked to digitize the documents by either scanning them or having people read and manually type them in. These are not ideal solutions, as modern computers are often not able to recognize the content due to faded ink and unrecognizable handwriting and fonts, and the effort is labor-intensive if people are reading and typing the documents. Human error is an additional risk to having people manually digitize the documents. A more ideal, and hopefully more accurate, solution would be to create a model to help computers better recognize the document content, and then have people verify the computer's interpretation. This would result in a more thorough and accurate digitizing process, while also decreasing the time people would need to spend on this effort.

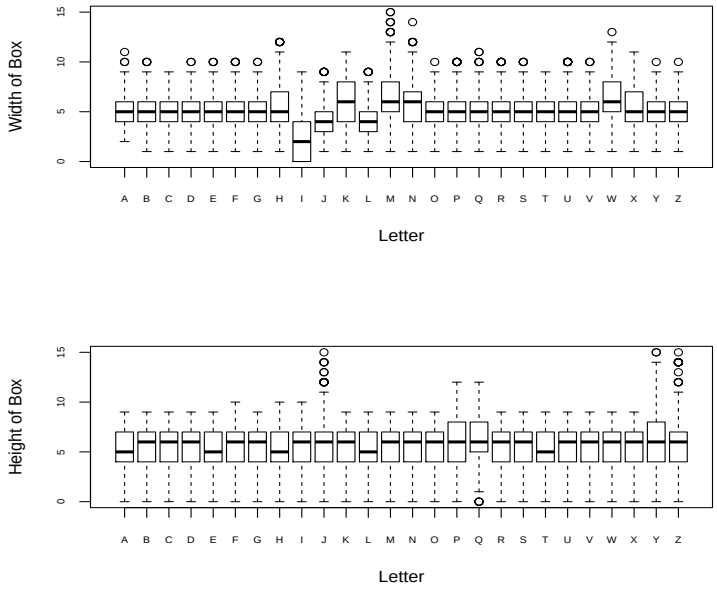
The purpose of this analysis is to build a model to feed to computers in an effort to improve their ability to recognize the hand-written text. The model will specifically work to increase recognition of individual letters based on certain characteristics of the letters. This is a prediction problem, because we intend for computers to use this model to identify letters based on their characteristics in hand-written historical documents. We proceed by first exploring the data, then building a model (and verifying its fit) that we determine is useful for the prediction purposes of this analysis.

2 Data Summary

The dataset used in this analysis includes observations of 16 different characteristics of writing in historical documents, along with the "person-verified" letter that matches each of those 16 characteristic observations. The characteristics recorded for each letter include: horizontal position of the box, vertical position of the box, width of box, height of the box, total number of pixels, mean x of pixels, mean y of pixels, mean x variance, mean y variance, mean x y correlation, mean of x*x*y, mean of x*y*y, mean edge count left to right, mean edge count bottom to top, correlation of x-ege with y, and correlation of y-ege with x.

There are 19,999 sets of observations in this dataset. The response variable in this analysis is the "person-verified" letter (A-Z), a categorical variable, which is not accounted for in simple linear regression. The linear model assumptions are also much more difficult to examine in this situation. We explored and observed some of the trends in the data using side-by-side boxplots for each of the variables against the letters. In Figure 1, we display a couple of these box plots.

Fig. 1. Boxplots of Some of the Explanatory Variables Against the Letters



The boxplots in Figure 1 show the distributions of the widths and heights of the boxes enclosing the letter regions. As expected letters I, J, and L had narrower boxes because they are relatively "skinny" letters. In contrast M, N and W are "wider" letters, and thus have a larger width box. The height of the box does not give us as much information as the width and some of the other explanatory variables, since the heights of boxes are relatively similar for each letter. We explored the other 14 variables the same way.

After exploring this data and its idiosyncrasies, we proceed to an exploration of analysis options. Because the response variable is categorical, we have to rethink the way we analyze the data. A categorical analysis option is considered in the following section.

3 Model

We choose to use a random forest method for our data analysis. Random forest is a powerful technique for prediction, using multi-level binary partitioning to predict a classification based on its explanatory values. Since the goal of our study is to predict letters given explanatory characteristics of the letters, random forest is an appropriate method to use. Before describing how random forests work, we first introduce the ideas of classification trees and bagging, which help both to explain the random forest method and to justify the reason for using the random forest method for our analysis.

Classification trees divide the prediction space into T non-overlapping spaces, or partitions, where at each level of the tree, the variable and the cut off point for the binary split are chosen such that we see the most reduction in the error. This binary partition is then replicated within each of the two previously assigned regions, and is repeated until a further split will not significantly reduce the error. Classification trees are used for prediction of data where the response variable is categorical. The model for classification trees is written as follows:

$$\hat{y}(\mathbf{x_o}) = \sum_{t=1}^{T} (arg\max_{c} \pi_c) I(\mathbf{x_0} \in \mathbf{R}_t), \text{ where } \pi_c = \text{Proportion of } \{y(\mathbf{x_i}) : \mathbf{x_i} \in \mathbf{R}_t\}$$

The function above gives the predicted values of the response, given input of the explanatory values, based on what the model indicates is the most likely classification. One of the greatest advantages of trees is that

they are non-linear, which is important for our dataset, given that it is difficult to test linear assumptions for categorical responses. However, we find that classification trees have a tendency to overfit (despite the option to build a tuning parameter to reduce the risk of overfitting) and have high variance, which both undermine their predictive performance. We consider bagging as a possible remedy to this high-variance and overfitting issue.

Bagging, or bootstrap aggregating, decreases the high variance issue with classification trees, which allows for overfitting and results in low bias. At each of $b = 1, ..., B$ iteration of the random forest algorithm, it takes a bootstrapped sample of the data with replacement, then builds a tree. The results of the trees from each iteration are aggregated, thereby reducing the variance. Additionally, overfitting the trees at each iteration in bagging does not reduce the predictive power of the model because these results are aggregated. The observations not included in the bootstrap sample can be used to calculate the out-of-bag error (OOB), which indicates the predictive abilities of the model. The predictions for this method are found by:

$$\hat{y}(\mathbf{x_o}) = \text{mode } (\hat{y}^1(\mathbf{x_0}), ..., \hat{y}^B(\mathbf{x_0})) \ ,$$

or the most common classification in the B iterations. While bagging reduces the variance and allows for overfitting, an improvement from the classification trees method, the variance is not reduced as much as it could be because the trees built in the iterations are highly correlated with each other.

To maximize our decrease in variance, and therefore our predictive abilities, we finally introduce the concept of random forests. Random forests are essentially a special case of bagging, where instead of considering all P variables for the partitioning at each split, only a randomly selected $m < P$ number of variables are considered at each split. This method solves the issue of high correlation between the trees and consequently reduces the variance even more (this is explained further in the following section). We therefore determine that this is the best method to use for our analysis. We note that the predictions for the random forest method are calculated the same way as the bagging method.

3.1 Model Specification

As indicated above, the random forest algorithm randomly considers $m < P$ number of variables at each split, where P is the total number of variables. This method of randomly considering m number of variables is used in order to decorrelate the trees. The process of decorrelating the trees decreases the variance more than bagging because taking the average of highly correlated quantities does not reduce the variance as much as averaging uncorrelated values. We use a tuning function for random forests to find the optimal m with respect to the Out-of-Bag (OOB) error estimate. We create an algorithm, where at each iteration, we estimate the OOB error for $m = i$, where $i = 1, 2, ..., 15$. The results vary each time this is run because the method is based on bootstrapping, or random draws. Because the optimal m varies each time this function is run, we replicate this process 500 times, and count the number of times that each value of m is chosen as the optimal value. We additionally estimate the average OOB error for each value of m.

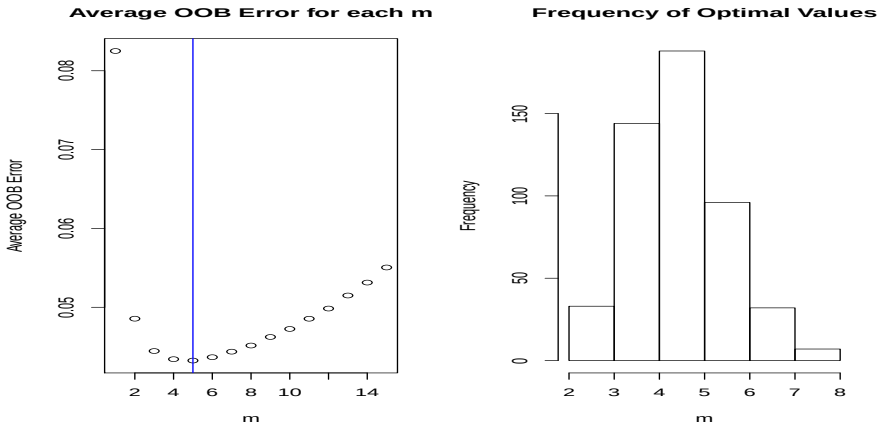
The two plots that represent these results can be found in Figure 2. As is shown in the plots, both the average OOB error values and the frequency that each value is chosen as optimal indicate that $m = 5$ is the optimal value. We note that only 6 values of m were ever chosen as optimal values in our 500 iterations ($m = 2, ..., 8$), and that $m = 5$ was chosen far more often than the others (with the exception of $m = 4$, which was fairly close). We proceed with building our model using this value, which means that at each binary partitioning of the trees in our random forest method, 5 variables will be randomly considered for the split.

3.2 Model Justification

As indicated, this model is a powerful prediction tool for classification analysis, resulting in more precise and less bias predictions than other methods discussed above. We additionally note that this model does not have any specific assumptions that need to be explained or justified. We therefore build our random forest model, keeping in mind that this will hopefully be used to help computers predict hand-written letters based on their characteristics.

We estimate the model fit by extracting the misclassification rate of our model. We additionally estimate the predictive power of our model by extracting our out-of-bag (OOB) error estimate. We note that both of

Fig. 2. Choosing the Optimal Value of m



these error rates are the same, indicating that the model performs as well out-of-sample as it does within sample. As indicated by the misclassification estimate, we find that the fitted values are only misclassified 3.3% of the time, and that out-of-bag values are also misclassified 3.3% of the time. This indicates that our model classifies letters correctly 96.7% of the time, both for in-sample and out-of-sample predictions. We determine that this is a well-fit model with high predictive power, and proceed to evaluating the results.

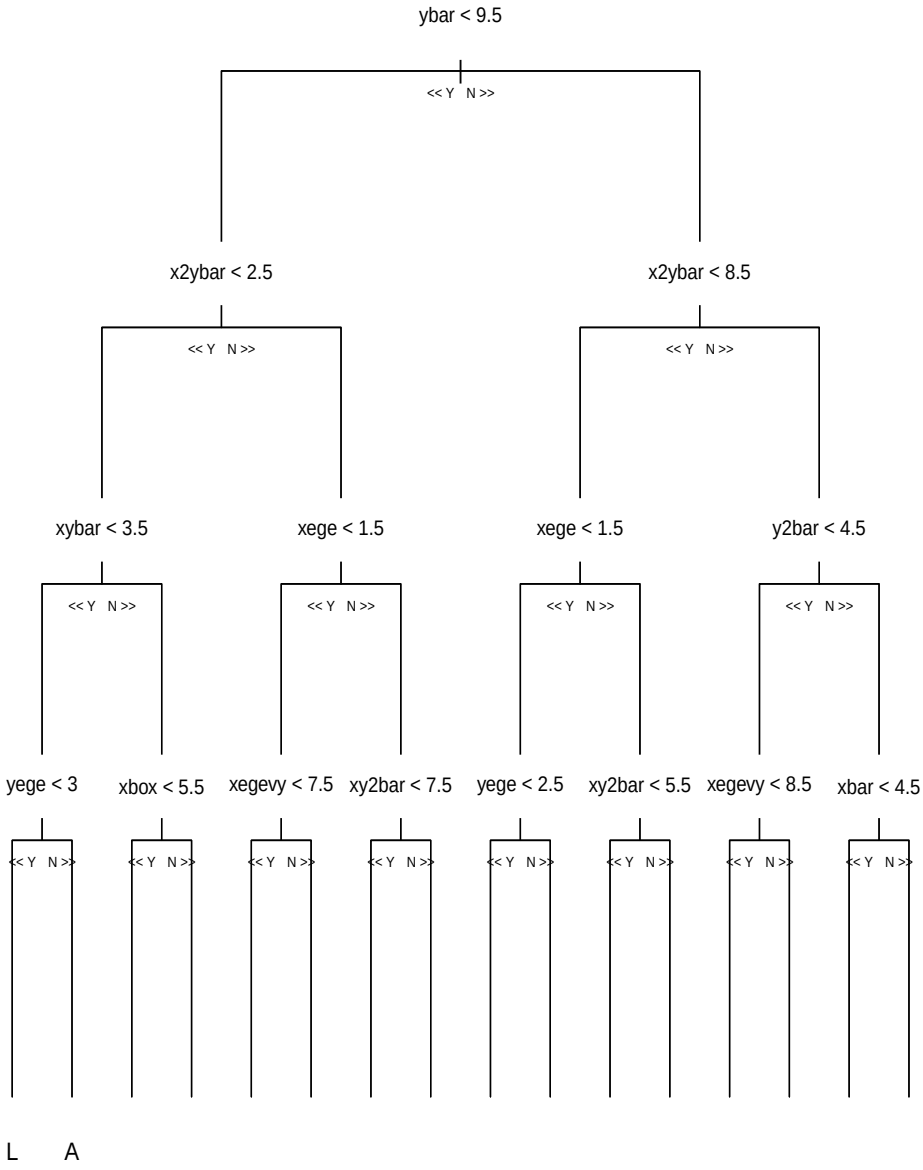
4 Results

As described before, the random forest method builds multiple trees. Showing each tree created would be imprudent, so we show one tree in Figure 3 to give a visual representation of the process. We also only include the first four splits of this tree, but note that there were many more past the point we display. The tree shows which variable is being considered to determine classification at each split, and the cutoff value is also displayed. Notice near the bottom of the tree and towards the left side are the letters L and A. This means, based on the loss function, it didn't make sense to add more branches to this part of the tree, so letters that met all the criteria at those points were assigned a letter L or A. There are no letters identified for the other branches because further splits were made before classification, which are not shown here as indicated above.

Table 4 gives us the actual letters as the rows and the predicted letters as the columns, and indicates the frequency that the actual letter was predicted to be each letter of the alphabet. As expected, the vast majority of letters were predicted correctly. In Table 4, we see that there are very low classification error rates. The letters predicted with the most success were A, M and W. Letters that have more distinct characteristics are easier to predict. For instance, in the case of M and W, there may be a low classification error rate because they are both relatively wide compared to other letters making them more distinguishable. The letters with the highest error rates are H, F and I. The letter H was most commonly misclassified as a K or an R, likely because of their similar height, width, and stroke patterns. The letter F was most commonly misclassified as a P, unsurprisingly because if the two horizontal lines in the letter F were connected then it would essentially be a P. The letter I was most commonly misclassified as a J, again unsurprising as these two are very similar and it would be easy to misjudge the horizontal line on the bottom of I as a curve for a J. We find that the class errors for each of the letters are all very low, and that overall the random forest model does well to predict letters, which is the goal of the analysis.

Scientists using the model with different data would be able to run the letter characteristics through the random forest and receive the predicted letters as the output. The predicted values for an unknown letter given the data represent the most commonly predicted letter across all trees. We again note that the model predicts just as well out-of-sample as it does in-sample, so we are confident with the model's predictive abilities with new data.

Fig. 3. This shows one of the trees in the random forest



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	class.error
A	705	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	2	0	0	0	0	0	2	0	0.010
B	0	652	0	0	2	0	2	2	0	0	0	0	1	1	0	0	0	5	0	0	1	6	0	2	0	0	0.033
C	0	0	644	0	5	0	6	2	0	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0.026
D	1	2	0	705	0	0	1	4	0	0	0	0	0	7	5	0	1	1	0	0	0	0	0	1	0	0	0.032
E	0	1	1	0	662	1	9	0	0	0	3	2	0	0	0	0	1	1	4	0	0	0	0	4	0	2	0.042
F	0	6	0	4	2	668	0	1	0	1	2	0	0	1	0	8	1	0	2	6	0	1	1	0	1	0	0.052
G	1	3	2	6	2	0	670	1	0	0	0	0	1	0	2	0	4	0	0	0	0	1	1	0	0	0	0.035
H	0	4	0	10	1	0	2	600	0	1	20	0	3	0	3	1	2	12	1	0	1	0	0	0	0	1	0.094
I	0	2	0	1	0	5	0	0	626	17	0	0	0	0	0	3	0	0	2	2	0	0	0	1	0	1	0.052
J	1	1	0	0	0	3	0	3	19	647	1	1	0	1	0	0	1	0	1	0	0	0	0	0	0	1	0.049
K	0	1	0	0	2	0	0	7	0	0	629	0	0	0	0	0	0	13	0	0	2	0	0	7	0	0	0.048
L	0	1	1	0	3	0	3	0	0	0	1	669	0	0	0	0	3	2	1	0	0	0	0	1	0	0	0.023
M	0	0	0	0	0	0	2	0	0	0	1	0	708	1	0	1	0	0	0	0	0	0	3	0	0	0	0.011
N	0	0	0	7	0	0	0	3	0	0	0	0	3	691	4	0	0	4	0	0	1	1	0	0	0	0	0.032
O	0	3	1	11	0	0	0	0	0	0	0	0	0	1	665	1	5	2	0	0	1	0	0	0	0	0	0.036
P	0	3	0	1	0	19	1	3	0	0	0	1	0	0	1	688	1	0	0	0	0	1	0	0	2	0	0.046
Q	1	2	0	2	0	0	1	0	0	0	0	0	0	0	13	2	677	2	0	0	0	0	0	0	1	0	0.034
R	0	10	0	2	0	0	0	2	0	1	9	0	1	4	0	1	1	628	0	0	0	0	0	1	0	0	0.048
S	0	8	0	1	2	1	0	2	0	1	0	0	0	0	0	0	1	1	658	0	0	0	0	0	0	0	0.025
T	0	0	2	1	0	3	0	0	0	0	2	0	0	0	0	0	1	0	2	695	0	1	0	0	5	1	0.025
U	0	0	0	0	0	0	0	2	0	0	1	0	7	1	1	0	0	0	0	0	723	2	0	0	0	0	0.019
V	0	13	0	0	0	1	2	0	0	0	0	0	2	0	0	3	0	0	0	0	0	665	2	0	3	0	0.038
W	0	0	0	0	0	0	1	0	0	0	0	0	4	1	2	0	0	0	0	0	1	1	674	0	0	0	0.015
X	0	2	0	2	3	0	0	2	0	0	6	0	0	0	0	0	0	1	0	0	0	0	0	685	1	0	0.024
Y	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	5	4	3	0	0	697	0	0.020
Z	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	9	1	0	0	0	0	0	0	0	658	0.019

Table 1. Confusion matrix with the actual letters as the rows and the predicted letters as the columns.

5 Conclusion

This model is very useful in the digitization efforts for hand-written historical documents. The predictive power is high, and we find that this will likely reduce the error that is inherent in the current digitizing efforts as well as reduce the time people will have to spend working on digitizing. With this predictive random forest model, computers will be better able to recognize letters in hand-written documents, and human effort will only be needed to verify the computer's predictions. We do however, acknowledge that the data used to build this model was only from a single historical record, which possibly makes this model less generalizable than we intended for it to be. We hope that future efforts will be made to expand the dataset and verify/update the model based on the updated findings. In future analyses it may be worthwhile to incorporate punctuation, and upper or lower case in the model. Our model does not try to identify any special characters or upper vs. lower case distinctions. Also, in the past there was a higher propensity to write in cursive. Cursive vs. non-cursive letters would most likely have major differences, so these should also be compared in future studies.