Project 3: *"Brought to you by the letter ..."*

Andrew Bernath, Heather Kitada, Ethan Edwards

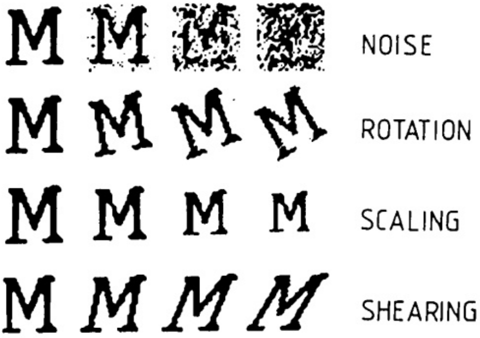
Oregon State University

June 4, 2014

Contents

Question of Interest

Classify an image of a letter to one of the 26 capital letters in the English alphabet.



http://imagebank.osa.org/getImage.xqy?img=dTcqLmxhcmdlLGFvLTIzLTEwLTE1MDktZzAxMA

Data Set Information

- All 26 uppercase English letters
- 20 fonts for each letter
- Randomly distorted
  - File of 20,000 unique observations
- Each observation converted into 16 primitive numerical attributes

16 Variables Used:

1. **lettr**: True capital letter (26 values from A to Z)
2. **x-box**: Horizontal position of box (integer)
3. **y-box**: Vertical position of box (integer)
4. **width**: Width of box (integer)
5. **high**: Height of box (integer)
6. **onpix**: Total number on pixels (integer)
7. **x-bar**: Mean x of on pixels in box (integer)
8. **y-bar**: Mean y of on pixels in box (integer)
9. **x2bar**: Mean x variance (integer)
10. **y2bar**: Mean y variance (integer)
11. **xybar**: Mean xy correlation (integer)
12. **x2ybr**: Mean of xxy (integer)
13. **xy2br**: Mean of xyy (integer)
14. **x-ege**: Mean edge count left to right (integer)
15. **xegvy**: Correlation of x-ege with y (integer)
16. **y-ege**: Mean edge count bottom to top (integer)

Description of Methods

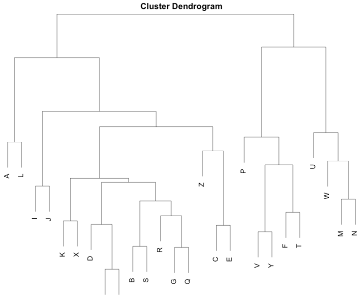
Algorithms for:

1. Logistic Regression Binary Search Tree (BST)
2. Decision Trees for Classification
   1. CART Method
   2. Bag Method

Logistic Regression BST Algorithm

Preparing Binary Tree (Using Learning Set):

1. Summarize by unique letter (average over observations from a given letter for each of the metrics)
2. Find distance between letters (uses Euclidean distance)
3. Use hclust() with "complete" method to create dendrogram



Logistic Regression BST Algorithm

Traversing Binary Tree with Logistic Regression Models:

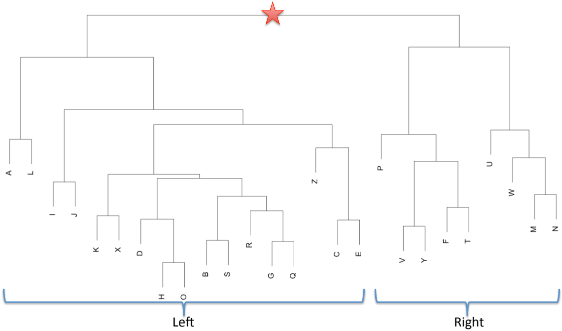
1. Subset letters are to the left and right of current intersection location. Right letters $= 1$, Left letters $= 0$
2. Create logistic regression model for probability of right (uses all 15 explanatory variables)
3. Evaluate logistic regression model with new covariates from observation in validation set.

$$\begin{cases} \text{move right} & : \text{if } \hat{\pi} \geq 0.5 \\ \text{move left} & : \text{if } \hat{\pi} < 0.5 \end{cases}$$

4. Keep track of path traversed
5. Repeat steps 1-4 until you arrived at an end node, which is the predicted letter

Logistic Regression BST Algorithm Example

New observation: (T, 2, 6, 3, 4, 2, 7, 12, 2, 7, 7, 11, 8, 1,11, 1, 8)

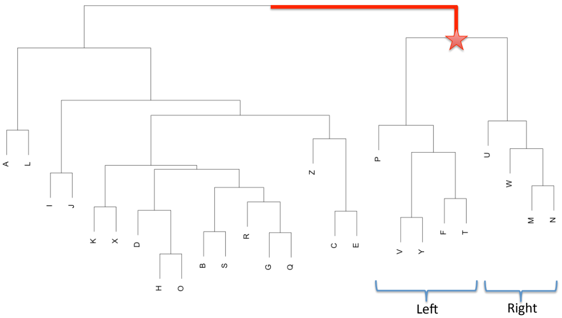


$$log(\frac{\pi_i}{1-\pi_i}) = -.5 + .32x_1 - .08x_2 + .07x_3 - .1x_4 + .11x_5 - .05x_6 + .41x_7 - .09x_8 - .3x_9 - .05x_{10} + .54x_{11} - .68x_{12} + .56x_{13} + .23x_{14} - .58x_{15} - .24x_{16} \rightarrow \hat{\pi} = 0.929$$

Move right!

Logistic Regression BST Algorithm Example

New observation: (T, 2, 6, 3, 4, 2, 7, 12, 2, 7, 7, 11, 8, 1,11, 1, 8)

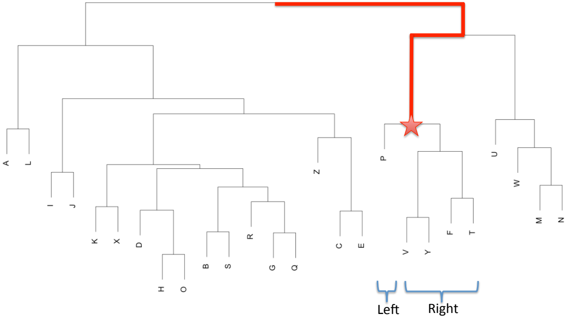


$$log(\frac{\pi_i}{1-\pi_i}) = 4.12 - .37x_1 + .15x_2 + .83x_3 - 1.07x_4 + .3x_5 - .64x_6 + .23x_7 + 1.17x_8 + .58x_9 - .39x_{10} - .83x_{11} + .88x_{12} + 1.87x_{13} - .51x_{14} - 2x_{15} - .57x_{16} \rightarrow \hat{\pi} = 0.0007$$

Move left!

Logistic Regression BST Algorithm Example

New observation: (T, 2, 6, 3, 4, 2, 7, 12, 2, 7, 7, 11, 8, 1,11, 1, 8)

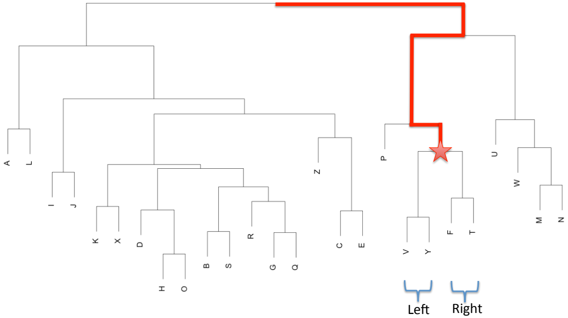


$$log(\frac{\pi_i}{1-\pi_i}) = -23.41 + .16x_1 + .17x_2 + .04x_3 - .25x_4 - .49x_5 + .38x_6 + .67x_7 - .65x_8 + .69x_9 + .23x_{10} + .91x_{11} + 1.79x_{12} + .36x_{13} - .1x_{14} + .07x_{15} - .29x_{16} \rightarrow \hat{\pi} = 0.999$$

Move right!

Logistic Regression BST Algorithm Example

New observation: (T, 2, 6, 3, 4, 2, 7, 12, 2, 7, 7, 11, 8, 1,11, 1, 8)

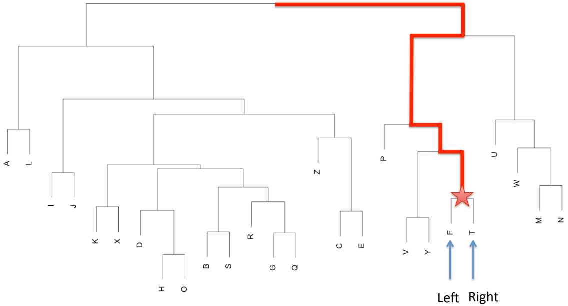


$$log(\frac{\pi_i}{1-\pi_i}) = -13.86 - .61x_1 + .5x_2 - .96x_3 - .49x_4 + 1.57x_5 + .57x_6 + 1.64x_7 + .69x_8 + 1.56x_9 + .85x_{10} - 1.71x_{11} + .32x_{12} - .65x_{13} - .96x_{14} - .55x_{15} + .58x_{16} \rightarrow \hat{\pi} = 0.991$$

Move right!

Logistic Regression BST Algorithm Example

New observation: (T, 2, 6, 3, 4, 2, 7, 12, 2, 7, 7, 11, 8, 1,11, 1, 8)

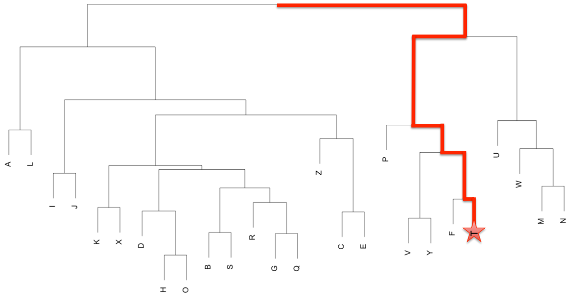


$$log(\frac{\pi_i}{1-\pi_i}) = -33.85 + .99x_1 + .77x_2 - .59x_3 - 1.36x_4 - .04x_5 + 1.5x_6 + 2.41x_7 + 1.22x_8 + 3.35x_9 - 1.96x_{10} - .87x_{11} + 1.61x_{12} + .33x_{13} + .66x_{14} - 1.25x_{15} - 1.32x_{16} \rightarrow \hat{\pi} = 0.999$$

Move right! and STOP

Logistic Regression BST Algorithm Example

New observation: (T, 2, 6, 3, 4, 2, 7, 12, 2, 7, 7, 11, 8, 1,11, 1, 8)



Prediction: T
Conclusion: Correctly classified! Yay!

Decision Tree Algorithm

Constructing Decision Tree (Using Learning Set):

1. All training set observations are lumped into a single node
2. The majority class - which class of letter has the most observations in the active node - is identified.
3. The Gini index is calculated for the active node.
    1. For every covariate at every possible splt point the Gini index is calculated for the two new created nodes after the considered splot.
    2. A weighted average is taken on the two indices.
    3. The coviariate/split point combination that produces the largest (Original Gini Index - Sum(Split Gini Indices)) is chosen as the split criteria.
4. The split is created creating two new nodes.
5. Steps 2 through 4 are repeated for each new node, up to a certain threshold.

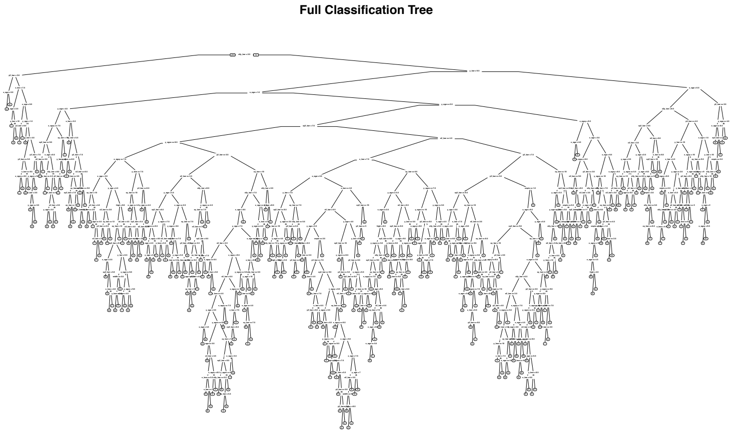
What is a Gini Impurity Index (GII)?

- The Gini index is a number that represents the "impurity" in a node, i.e. the amount of mixing of classes present
- A pure node would be one consisting of only a single class, then Gini index = 0
- A node with equal amounts of every class would be perfectly impure, and the Gini would be at maximum (no upper bound)

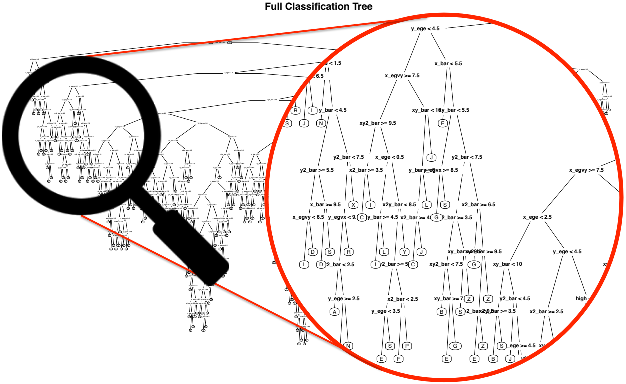
Traversing Decision Tree

1. A new observation is introduced.
2. The first decision point - i.e., split point/covariate combination - is reached. If the covariate for the new observation is less than the split point, it goes left; if it is greater, it goes right.
3. Step 1 is repeated until a terminal node is reached, and a class is assigned.

Full CART Decision Tree

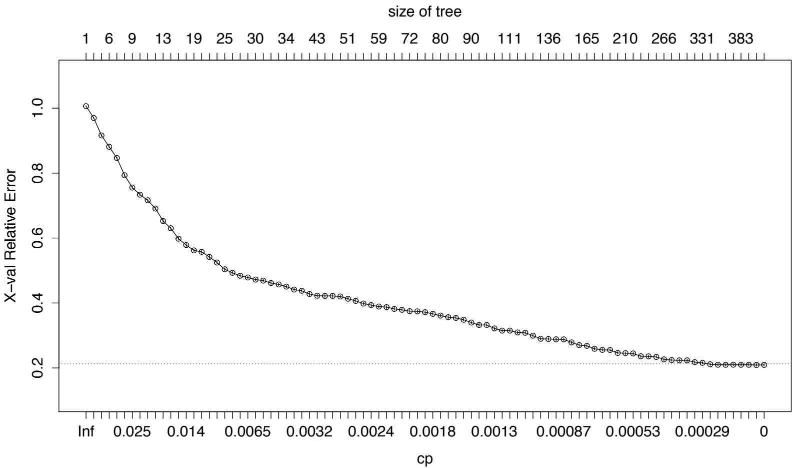


Full CART Decision Tree Zoom In

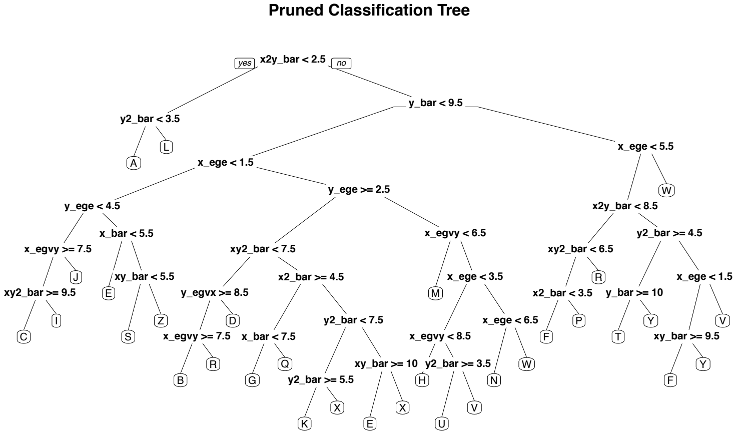


After tree is created it requires "pruning" to get rid of repeated nodes.

How should we prune?



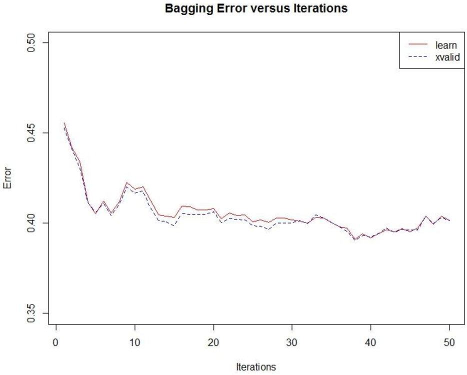
Pruned CART Decision Tree



CART vs BAG Methods

- CART model is based on using a single tree for each of the predictions made.
  - Fails to classify 7 classes
- BAG model is based on aggregation (bootstrap) of votes from all the trees used in the model.
  - performs better (it predicts all classes, even if not perfectly)

Bagging Error Plot



Summary of Findings

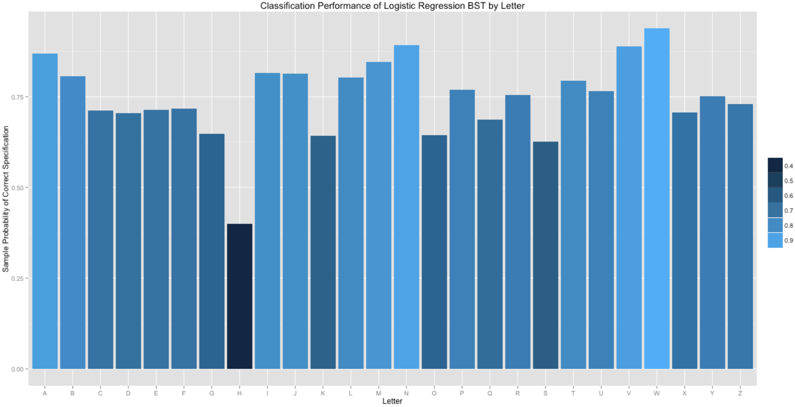
Findings for:

1. Logistic Regression BST Confusion Matrix
2. Decision Trees for Classification
    1. CART Method Confusion Matrix
    2. Bag Method Confusion Matrix

Overall Findings

1. Logistic Regression BST : 74.8% Correct Specification Overall
   - Highest Correct Classification: **W** with 94%
   - Lowest Correct Classification: **H** with 40%
2. CART Method: 47.1% Correct Specification Overall
   - Highest Correct Classification: **I** with 78%
   - Lowest Correct Classification: **E,F,K,O,R,S,Y** with 0%
3. Bag Method: 60.6% Correct Specification Overall
   - Highest Correct Classification: **V** with 82%
   - Lowest Correct Classification: **S** with 22%

Logistic Regression BST Distribution of Specification



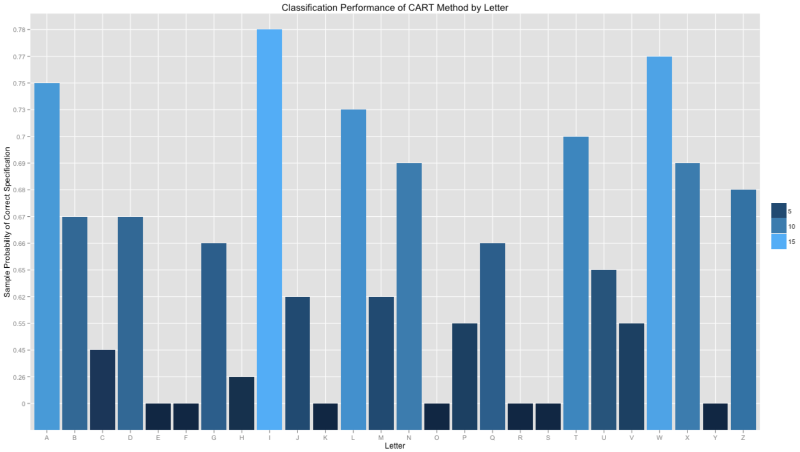
Logistic Regression BST Confusion Matrix

Logistic Regression Binary Search Tree Confusion Matrix

True Letter (columns) / Chosen Letter (rows)

	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
A	0.87	0	0	0	0	0	0	0	0	0	0	0	0.06	0	0	0	0.03	0	0	0	0	0	0	0	0	0
B	0	0.81	0.01	0.05	0.03	0.04	0.01	0.01	0.04	0	0.01	0.01	0	0	0	0.01	0.02	0.03	0.05	0.02	0	0	0	0.04	0	0.02
C	0	0	0.71	0	0.01	0	0.1	0	0	0	0.04	0.01	0	0	0	0	0.02	0	0	0	0	0	0	0	0	0
D	0	0.02	0	0.7	0	0.04	0.01	0.03	0	0.01	0.01	0	0	0.01	0.01	0.05	0	0.03	0.02	0.01	0	0	0	0.02	0.01	0
E	0	0	0.03	0	0.71	0.02	0.02	0	0	0.01	0	0.06	0	0	0	0	0.02	0	0	0.03	0	0	0	0.02	0.02	0.03
F	0	0	0	0	0	0.72	0	0.04	0	0.01	0	0	0	0	0.01	0.07	0	0	0	0.01	0	0	0	0	0.05	0
G	0	0.02	0.04	0	0.08	0.01	0.65	0.03	0	0	0.07	0.04	0	0	0.05	0.04	0.04	0.01	0.01	0.01	0	0.01	0	0.02	0	0
H	0	0.02	0.04	0.02	0	0	0	0.4	0	0.02	0.01	0	0.04	0.03	0.1	0	0	0.03	0	0	0.04	0.01	0.01	0	0	0
I	0	0	0	0	0	0.02	0	0	0.82	0.03	0	0	0	0	0	0	0.01	0	0	0.03	0	0	0	0	0	0
J	0.01	0	0	0	0	0	0	0	0.02	0.81	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0.03
K	0	0	0.09	0	0.03	0	0.01	0.02	0	0	0.64	0	0	0.02	0	0	0	0.05	0	0.01	0.03	0	0	0.02	0	0
L	0	0	0	0	0	0	0	0	0.01	0	0	0.8	0	0	0	0	0.01	0	0.05	0	0	0	0	0	0	0
M	0.01	0	0	0	0	0	0	0.01	0	0	0	0	0.85	0	0	0	0	0.01	0	0	0.01	0	0.03	0	0	0
N	0.01	0	0	0.02	0	0.01	0	0.17	0	0	0.01	0	0.01	0.89	0	0	0	0.02	0	0	0.02	0	0.01	0	0	0
O	0	0.01	0.03	0.02	0	0	0	0.07	0	0.02	0	0	0	0.01	0.64	0.01	0.07	0.02	0.04	0	0.08	0.01	0	0	0	0
P	0	0	0	0.09	0	0.04	0	0.01	0	0	0	0	0	0.01	0.02	0.77	0	0	0	0	0	0	0	0	0.01	0
Q	0.01	0.01	0	0	0.01	0	0.08	0.04	0.02	0.03	0.01	0.04	0	0	0.05	0	0.69	0.02	0.02	0	0	0	0	0.02	0.02	0.01
R	0	0.04	0	0.02	0.03	0	0.01	0.07	0	0	0.07	0	0.01	0	0.03	0	0	0.75	0.03	0.01	0	0.03	0	0	0	0.02
S	0.03	0.03	0.02	0.03	0.04	0.06	0.05	0.01	0.06	0.05	0	0.02	0	0	0	0.01	0.04	0	0.63	0.03	0	0	0	0.07	0.04	0.16
T	0	0	0.01	0	0	0.02	0	0.01	0	0	0	0	0	0	0.01	0	0	0	0	0.79	0.01	0	0	0.01	0.05	0
U	0.02	0	0.01	0.04	0	0	0	0.02	0	0	0.08	0	0.01	0	0	0	0	0	0	0	0.77	0	0	0	0	0
V	0.02	0.02	0	0	0	0.01	0.03	0.04	0	0	0.01	0	0	0	0.01	0.01	0.02	0	0	0.01	0.01	0.89	0.01	0	0.04	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0.01	0.06	0	0	0	0	0	0.02	0.02	0.94	0	0	0
X	0	0.01	0	0	0.02	0.01	0.01	0	0.03	0	0.01	0.01	0	0	0	0	0.01	0.01	0.01	0	0	0	0	0.71	0	0
Y	0.02	0	0	0	0	0.01	0	0.02	0	0	0.02	0	0	0	0	0.02	0	0	0.01	0.01	0	0.02	0	0.06	0.75	0
Z	0	0.01	0	0	0.03	0.01	0	0	0	0	0	0	0	0	0	0	0.03	0	0.11	0.02	0	0	0	0	0	0.73

CART Method Distribution of Specification



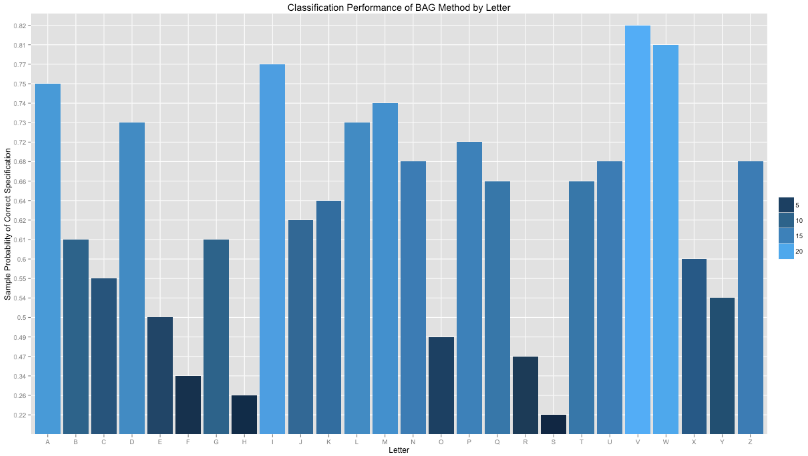
CART Method Confusion Matrix

CART Confusion Matrix

True Letter (columns); Chosen Letter (rows)

	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
A	0.75	0	0	0	0	0	0	0	0	0	0	0	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0.67	0	0.17	0	0.04	0.03	0.08	0.09	0.08	0.04	0.01	0	0.01	0.05	0.05	0.03	0.29	0.06	0	0	0.01	0.01	0.05	0	0
C	0	0	0.45	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0.02	0.16	0.01	0.67	0	0.15	0.04	0.16	0.02	0.07	0.01	0	0.1	0.1	0.18	0.23	0.02	0.17	0.2	0.05	0.03	0.02	0.02	0.16	0.13	0.12
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0.02	0.04	0.25	0.04	0.18	0.02	0.66	0.16	0	0	0.15	0.02	0.04	0.01	0.36	0.05	0.07	0.13	0.03	0.04	0.12	0.04	0.04	0.05	0	0.05
H	0	0	0	0.08	0	0	0	0.26	0	0.01	0.11	0	0	0	0.07	0	0	0	0	0	0.01	0	0	0.01	0	0
I	0	0.01	0	0	0.01	0.03	0	0	0.78	0.02	0	0.01	0	0	0	0.09	0	0	0.05	0	0	0	0	0	0.05	0
J	0	0	0.05	0	0	0	0	0	0.02	0.62	0	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0.03	0	0	0.02	0	0	0	0.01	0.01	0.1	0.02	0.73	0.02	0.04	0	0	0.01	0.08	0.08	0	0	0	0	0.04	0	0.02
M	0.02	0	0	0	0	0	0	0	0	0	0	0	0.62	0	0	0	0	0	0	0	0.01	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0.05	0	0.01	0.69	0	0	0	0	0	0	0.08	0.02	0.06	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0.02	0	0.01	0.3	0	0.03	0.02	0.02	0	0	0	0.03	0	0.55	0.03	0.16	0.01	0.06	0	0.01	0.01	0	0.02	0.02
Q	0.03	0.04	0.03	0.01	0.05	0	0.07	0.04	0.02	0.03	0	0.01	0.03	0	0.32	0	0.66	0.04	0.11	0.01	0.05	0.02	0.04	0	0.06	0.01
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0.01	0	0.01	0.23	0	0	0	0	0	0	0	0	0	0.03	0	0	0	0.7	0.01	0.03	0	0	0.39	0
U	0	0	0.02	0	0	0.02	0	0.01	0	0	0	0	0	0	0	0	0.01	0	0	0	0.65	0.25	0.01	0	0.02	0
V	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0.02	0	0	0.03	0	0	0.07	0	0.55	0.04	0	0.28	0
W	0	0	0	0	0	0.01	0.01	0	0	0	0	0	0.08	0.07	0.01	0	0.01	0	0	0	0.02	0.02	0.77	0	0.01	0
X	0.13	0.02	0.16	0.01	0.45	0	0.05	0.24	0.03	0.03	0.63	0.2	0.07	0.03	0.02	0	0.12	0.13	0.17	0.07	0.03	0.03	0	0.69	0.04	0.09
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z	0	0.06	0	0	0.27	0	0.13	0	0	0	0	0	0	0	0	0	0	0	0.28	0	0	0	0	0	0	0.68

BAG Method Distribution of Specification



BAG Method Confusion Matrix

BAG Confusion Matrix

True Letter (columns); Chosen Letter (rows)

	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
A	0.75	0	0	0	0	0	0	0	0	0	0	0	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0.61	0.02	0.12	0.03	0.05	0.02	0.03	0.08	0.09	0.02	0.01	0	0.01	0.04	0.06	0.05	0.08	0.05	0.01	0	0.01	0	0.04	0	0
C	0	0	0.55	0	0	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0.02	0.14	0	0.73	0	0.14	0.04	0.12	0.02	0.08	0.01	0	0.01	0.06	0.1	0.1	0.02	0.15	0.15	0.06	0.02	0	0	0.1	0.04	0.11
E	0.02	0	0	0	0.5	0	0.02	0	0	0.01	0	0.11	0	0	0	0	0.06	0	0.05	0	0	0	0	0.01	0	0.06
F	0	0	0	0	0	0.34	0	0.01	0	0	0	0	0	0	0	0.01	0	0	0	0.02	0	0.01	0	0	0	0
G	0.02	0	0.15	0.01	0.14	0.01	0.61	0.06	0	0	0.03	0.01	0	0	0.1	0.02	0.03	0.03	0.02	0.01	0.06	0	0	0	0	0.03
H	0	0	0	0	0	0	0	0.26	0	0.01	0	0	0	0	0.06	0	0	0	0	0	0	0	0	0	0	0
I	0	0.01	0.01	0	0.01	0.06	0	0	0.77	0.02	0	0.01	0	0	0	0.02	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0.01	0.62	0	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0.01	0	0.15	0	0.14	0	0.01	0.2	0	0	0.64	0.05	0	0.01	0	0	0.02	0.05	0.07	0.05	0.01	0	0	0.11	0.01	0
L	0.03	0	0	0.02	0	0	0	0.01	0.01	0.1	0.02	0.73	0.02	0.04	0	0	0.01	0.08	0.08	0	0	0	0	0.04	0	0.02
M	0.03	0.01	0	0	0	0	0.01	0.03	0	0	0.03	0	0.74	0	0	0	0	0.02	0	0	0.02	0	0.02	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0.05	0	0.01	0.68	0	0	0	0	0	0	0.07	0.01	0.07	0	0	0
O	0	0	0	0.01	0	0	0	0.07	0	0.01	0	0	0	0.02	0.49	0	0.02	0	0.01	0.01	0.02	0	0.02	0.01	0	0
P	0	0	0	0.01	0	0.14	0	0	0.01	0.01	0	0	0	0	0	0.72	0	0	0	0.03	0	0	0	0	0	0
Q	0.08	0.03	0.03	0.01	0.06	0	0.08	0.05	0.03	0.03	0.01	0.03	0.01	0.03	0.13	0	0.66	0.03	0.13	0.01	0.05	0.01	0.01	0	0.07	0.02
R	0	0.06	0.01	0.05	0	0.02	0.03	0.07	0	0	0.03	0.01	0	0	0.02	0	0.01	0.47	0.02	0	0	0	0	0	0.01	0.01
S	0	0	0	0	0	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0.22	0	0	0	0	0	0	0
T	0	0	0	0	0	0.09	0	0	0	0	0	0	0	0	0	0.02	0	0	0	0.66	0	0.01	0	0	0.15	0
U	0	0	0.02	0	0	0.02	0	0.01	0	0	0.01	0	0	0	0	0	0.03	0	0.01	0.01	0.68	0.03	0	0.01	0.04	0
V	0	0	0	0	0	0.01	0	0.01	0	0	0	0	0.01	0.05	0	0	0.04	0	0	0	0	0.82	0.07	0	0.1	0
W	0.01	0.04	0.01	0.02	0.01	0.01	0.03	0.03	0	0	0.03	0.01	0.18	0.09	0.05	0.03	0.01	0.04	0	0	0.06	0.05	0.81	0	0.03	0
X	0.03	0.02	0	0.01	0.05	0	0.03	0.04	0.02	0.02	0.05	0	0	0.01	0.01	0	0.03	0.06	0.09	0.04	0.01	0.03	0	0.6	0	0.06
Y	0	0	0.03	0	0.02	0.11	0	0.01	0	0	0.08	0	0	0	0	0.01	0	0	0	0.1	0	0.01	0	0.08	0.54	0.01
Z	0	0.07	0.01	0	0.03	0	0.08	0	0.02	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0.68

Discussion

- Logistic Regression Assumptions
- Decision Tree Assumptions
- Scalabilty

Usual Logistic Regression Assumptions

- The true conditional probabilities are a logistic function of the independent variables
- No important variables are omitted.
- No extraneous variables are included.
- The independent variables are measured without error.
- The observations are independent.
- The independent variables are not linear combinations of each other.

Source: IDRE UCLA (Institute for Digital Research and Education

Decision Tree Assumptions

CART assumptions:

1. Data are drawn independently from the same distribution.
2. Data is fixed and free of measurement error.
3. For classification trees the response must be discrete and the covariates are categorical, discrete, or can be discretized.

Bagging assumptions:

1. The underlying classifier has small bias
2. The underlying classifier is appropriate for the data

Scalability

- Both algorithms can be applied to bigger/different data sets
- However, more observations and/or more categories would increase run time
  - LRBST already takes a long time to run ($\sim$ 8 hrs) due to inefficient coding
  - Dicision trees run quickly because there are pre-packaged functions in R.
  - Might take more time to prune.

Future Work

- Make LRBST code more efficient by using object oriented programming aspects of R
- Decision trees with linear combinations of variables instead of splitting by variables one at a time
- Look at different letter cases, languages, symbols,...

Questions