Letter Classification

Josh Meyers, W. Zachary Horton

Brigham Young University

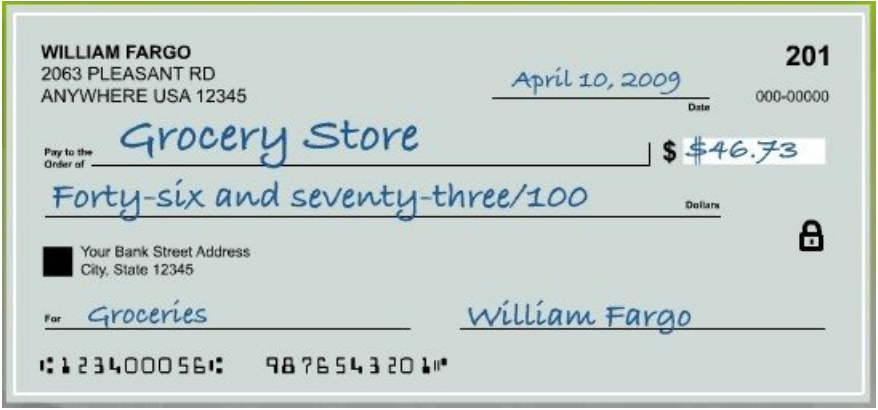
November 6, 2018

Introduction

Applications for Scanning Documents

- Family History (Indexing)
- Job Flyers
- and. . .

BANKING!



Note that accuracy is EXTREMELY important

Overview

GOAL: Build a System That Accurately Identifies Handwritten Letters

- Use data that contains handwritten letter properties
- Build the following models and tune using CV:
  - K-Nearest Neighbors (KNN)
  - Random Forest (RF)
  - Support Vector Machines (SVM)
  - Classification Tree Boosting (Boost)
  - Multinomial Logistic Regression (MREG)
- Make predictions using 10-fold out-of-sample methods
- Combine predictions using:
  - Majority Vote
  - Bayes Symphony
- Select best predicting method(s)

Data

20,000 Human-verified letters, 16 variables measured on each.

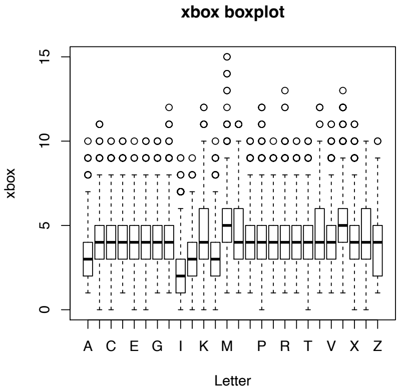
Letter Dataset Sample

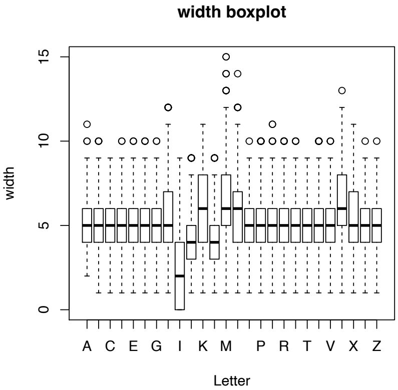
letter	xbox	ybox	width	high	pix	xbar	ybar	x2bar	...
I	5	12	3	7	2	10	5	5	
D	4	11	6	8	6	10	6	2	
				...					

- Variables represent pixel-based attributes
- Quantitative, hard to interpret
- Least represented: H (734)
- Most represented: U (813)

Data

Some variables are more useful than others for certain letters. . .





Multinomial Regression

- Similar to Logistic Regression
- Fit Using Numerical Techniques

$$Y_i \overset{iid.}{\sim} Mult(p_{1i}, \ldots, p_{Ki})$$

$$\log\left(\frac{p_{1i}}{p_{Ki}}\right) = \mathbf{x}`_i \beta_1$$

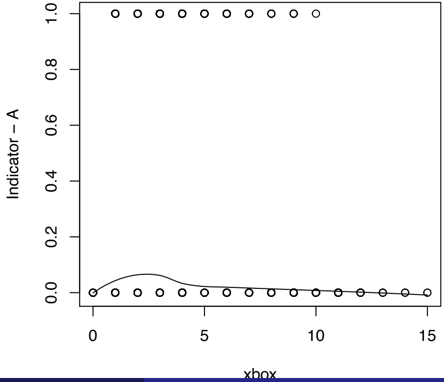
$$\vdots$$

$$\log\left(\frac{p_{(K-1)i}}{p_{Ki}}\right) = \mathbf{x}`_i \beta_{(K-1)}$$

- Predict using highest estimated probability (don't forget class K)

Multinomial Regression

- Used squared terms as well as linear terms
- Can't model interactions (too many)
- Used `nnet` package in `R`
- Standardized data (not automatic)



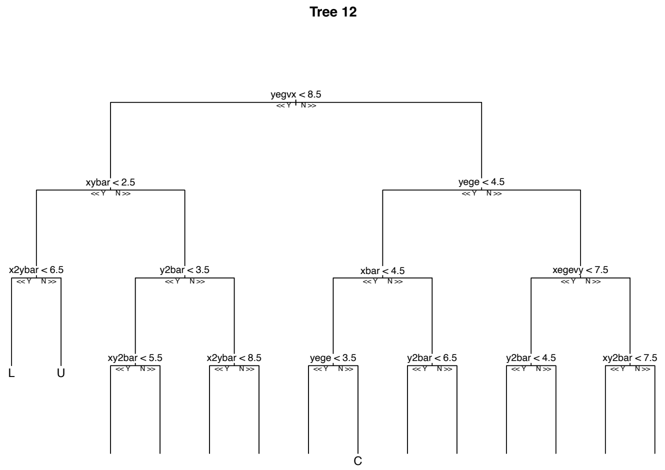
Random Forest

Algorithm: for b = 1, ..., B

1. Take a bootstrapped sample of size n (the total size of the data set).
2. At each split randomly consider 3 variables (chosen through cross validation).
3. Build a tree $\mathcal{T}_b$.

Using the data have each of the B trees classify each observation and assign a final classification based off of the majority vote.

Random Forest Example

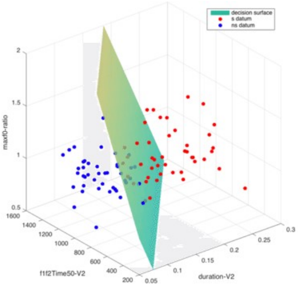


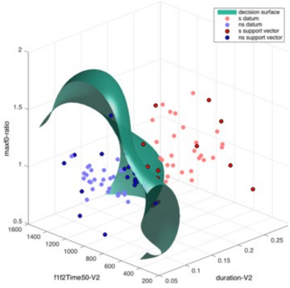
Support vector machines use kernels to separate the p-dimensional space into 2 distinct classes.

There are a few parameters that must be chosen:

- Kernel - Radial
- Cost - 13

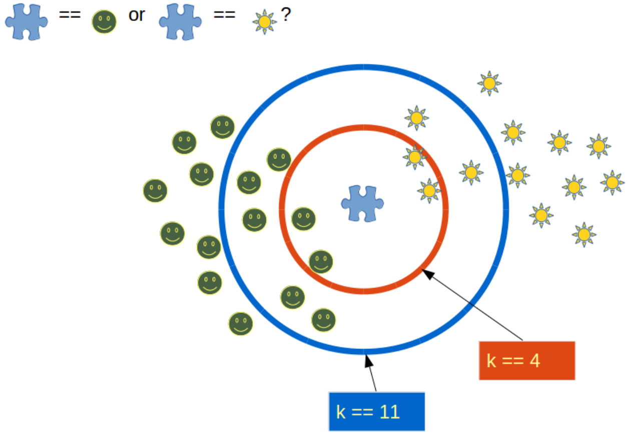
A one-to-one approch was used to classify each observation.





K Nearest Neighbors

- Majority vote between the nearest k data points
- Optimized for our data at k=3
- Standardized Euclidean distance



Boosting

Boosting fits trees to capture different aspect of the data. Instead of fitting only just the data, boosting iteratively assigns weights to observations as it explores and fits the data.
In the `gbm` package there are a couple of parameters that must be optimized.

- Loss Function - Multinomial
- Number of Iterations - 500
- Depth of each tree - 3
- Shrinkage parameter - .1
- Subsampling rate - .5

Combining Models

- Majority Rule
- Bayes Symphony
    - Based on Bayes Rule and Confusion Matrices
    - Ensembles Multiple Ensembles, hence "Symphony"
    - Up-weights models that predict certain letters well
    - Down-weights models the predict certain letters poorly
    - Cherry-picks the predictive power of multiple models

Bayes Symphony

DEFINITION :
Let $M1,\ldots,M5$ be our 5 prediction methods (models)
Let $m1,\ldots,m5$ be the realized predictions for the 5 methods
Let $\mathbf{M} = (M1 = m1, ..., M5 = m5)$ be the prediction set

$$P(\mathrm{A}|\mathbf{M}) \propto P(M1 = m1|A) * ... * P(M5 = m5|A) * P(A)$$

$$\vdots$$

$$P(\mathrm{Z}|\mathbf{M}) \propto P(M1 = m1|Z) * ... * P(M5 = m5|Z) * P(Z)$$

The final prediction is the letter with the highest posterior probability

Bayes Symphony

- Confusion matrices provide conditional probabilities
- Divide entries by column sums
- 5 total confusion matrices, one for each model

	Actual Letter		
Pred M1	A	B	C
A	P(M1=A|A)	P(M1=A|B)	P(M1=A|C)
B	P(M1=B|A)	P(M1=B|B)	P(M1=B|C)
C	P(M1=C|A)	P(M1=C|B)	P(M1=C|C)

Results

Method	Accuracy
Multinomial Logistic Regression	82.7
Random Forest	97.0
Support Vector Machine	97.4
KNN	95.7
Boosting	95.6
Majority Rule	97.5
Bayes Symphony	98.2

Multinomial Regression Results

$$
\begin{array}{c}
\\ B \\ C \\ G \\ K \\ Q \\ R
\end{array}
\begin{array}{cccccc}
B & C & G & K & Q & R \\
\end{array}
$$

$$
\begin{array}{c|cccccc}
 & B & C & G & K & Q & R \\
\hline
B & 629 & 0 & 13 & 2 & 6 & 32 \\
C & 1 & 570 & 38 & 44 & 1 & 0 \\
G & 5 & 43 & 538 & 12 & 46 & 11 \\
K & 7 & 18 & 4 & 574 & 0 & 43 \\
Q & 11 & 1 & 22 & 0 & 599 & 1 \\
R & 41 & 1 & 7 & 26 & 2 & 603
\end{array}
$$

Random Forest Results

$$\begin{array}{c c} & \begin{array}{ccccc} H & I & J & K & R \end{array} \\ \begin{array}{c} H \\ I \\ J \\ K \\ R \end{array} & \begin{bmatrix} 671 & 0 & 0 & 19 & 15 \\ 0 & 717 & 24 & 0 & 0 \\ 1 & 23 & 715 & 1 & 0 \\ 9 & 0 & 0 & 701 & 15 \\ 3 & 0 & 0 & 9 & 727 \end{bmatrix} \end{array}$$

Support Vector Machines Results

$$
\begin{array}{c|cccc}
 & D & H & I & J \\
\hline
D & 780 & 5 & 0 & 0 \\
H & 18 & 669 & 0 & 0 \\
I & 0 & 0 & 727 & 20 \\
J & 2 & 1 & 18 & 717
\end{array}
$$

Boosting Results

$$
\begin{array}{c c}
 & \begin{array}{cccccc} F & I & J & K & P & R \end{array} \\
\begin{array}{c} F \\ I \\ J \\ K \\ P \\ R \end{array} &
\begin{bmatrix}
734 & 3 & 3 & 1 & 11 & 0 \\
6 & 710 & 27 & 0 & 3 & 0 \\
2 & 24 & 706 & 0 & 1 & 0 \\
1 & 0 & 0 & 687 & 0 & 20 \\
16 & 1 & 0 & 0 & 765 & 0 \\
0 & 0 & 0 & 8 & 1 & 715
\end{bmatrix}
\end{array}
$$

K Nearest Neighbors Results

$$
\begin{array}{c}
\\
D \\ I \\ H \\ J \\ K \\ F \\ P
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} D & I & H & J & K & F & P \end{array} \\
\begin{bmatrix}
774 & 1 & 8 & 0 & 1 & 0 & 0 \\
1 & 725 & 0 & 26 & 0 & 1 & 0 \\
16 & 0 & 637 & 0 & 26 & 0 & 2 \\
0 & 26 & 1 & 707 & 0 & 3 & 0 \\
1 & 0 & 25 & 0 & 669 & 0 & 0 \\
3 & 3 & 0 & 1 & 0 & 716 & 28 \\
2 & 0 & 3 & 0 & 0 & 36 & 749
\end{bmatrix}
\end{array}
$$

Majority Rules Results

$$
\begin{array}{c}
\\ D \\ F \\ H \\ I \\ J \\ P
\end{array}
\begin{array}{cccccc}
D & F & H & I & J & P \\
\hline
\end{array}
$$

$$
\begin{array}{c|cccccc}
 & D & F & H & I & J & P \\
\hline
D & 790 & 1 & 15 & 1 & 0 & 1 \\
F & 0 & 752 & 0 & 5 & 1 & 17 \\
H & 3 & 1 & 672 & 0 & 1 & 2 \\
I & 0 & 0 & 0 & 724 & 21 & 0 \\
J & 0 & 0 & 0 & 21 & 716 & 0 \\
P & 0 & 8 & 0 & 1 & 0 & 773
\end{array}
$$

Bayes Symphony Machines Results

Perfect predictions for A's

$$
\begin{array}{c|cccccc}
 & D & F & H & I & J & P \\
\hline
D & 790 & 1 & 10 & 0 & 0 & 0 \\
F & 0 & 760 & 0 & 3 & 0 & 16 \\
H & 3 & 1 & 695 & 0 & 0 & 1 \\
I & 0 & 0 & 0 & 729 & 19 & 0 \\
J & 0 & 0 & 0 & 22 & 728 & 0 \\
P & 0 & 8 & 0 & 1 & 0 & 781
\end{array}
$$

Conclusion

Findings:

- Many models with Bayes Symphony has 98.2% accuracy
- To use: predict with all 5 and then symphony them together
- H and K were most commonly missed
- I and J were most commonly swapped
- MREG performed worst, but provided valuable information
- Random Forest and SVM predict best alone

Criticism:

- Boosting not totally optimized (we almost crashed hilbert)
- Use Mahalanobis Distance in KNN and MREG

Future Work:

- Use more models
- Simulation studies to test Symphony robustness
- Compare to Neural Network (and win!)