

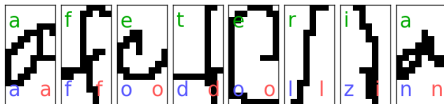
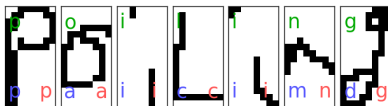
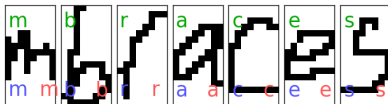
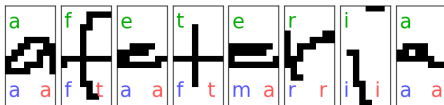
# Letter Classification

Josh Meyers, W. Zachary Horton

Brigham Young University

November 6, 2018

# Introduction



## Applications for Scanning Documents

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

## BANKING!

**WILLIAM FARGO**  
2063 PLEASANT RD  
ANYWHERE USA 12345

April 10, 2009 Date

**201**  
000-00000

Pay to the Order of Grocery Store \$ \$46.73

Forty-six and seventy-three/100 Dollars

Your Bank Street Address  
City, State 12345

For Groceries William Fargo

⑆ 123400056⑆ 9876543201⑆

Note that accuracy is EXTREMELY important

## 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)

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

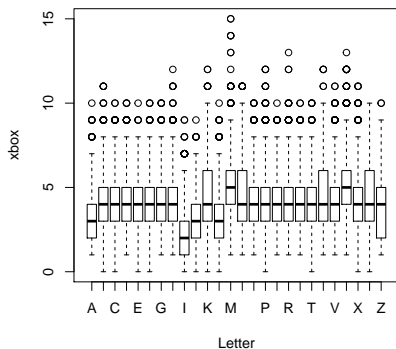
Letter Dataset Sample

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	
...									

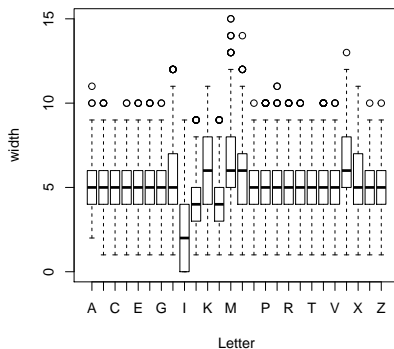
- Variables represent pixel-based attributes
- Quantitative, hard to interpret
- Least represented: H (734)
- Most represented: U (813)

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

**xbox boxplot**



**width boxplot**



# Multinomial Regression

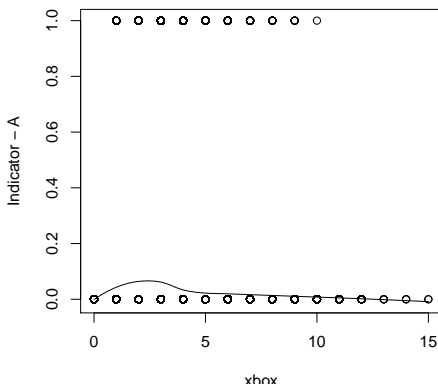
- Similar to Logistic Regression
- Fit Using Numerical Techniques

$$\begin{aligned} Y_i &\overset{iid.}{\sim} Mult(p_{1i}, \dots, p_{Ki}) \\ \log \left( \frac{p_{1i}}{p_{Ki}} \right) &= \mathbf{x}'_i \beta_1 \\ &\vdots \\ \log \left( \frac{p_{(K-1)i}}{p_{Ki}} \right) &= \mathbf{x}'_i \beta_{(K-1)} \end{aligned}$$

- 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

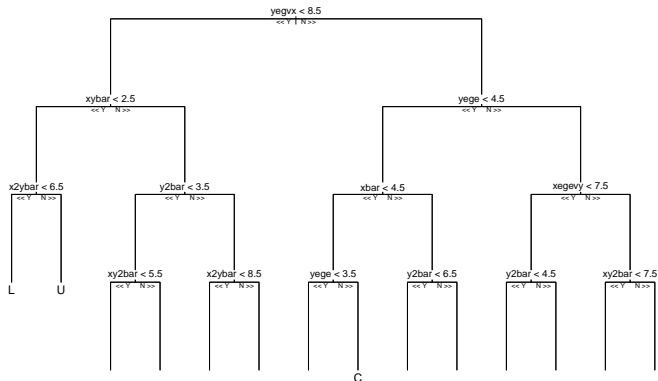
Algorithm: for  $b = 1, \dots, B$

- 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

Tree 12



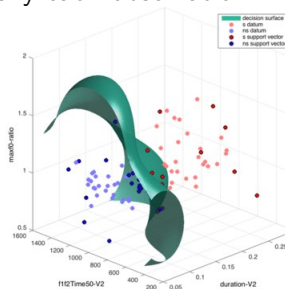
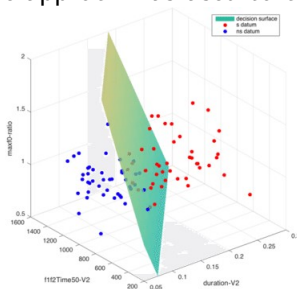
# Support Vector Machines

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

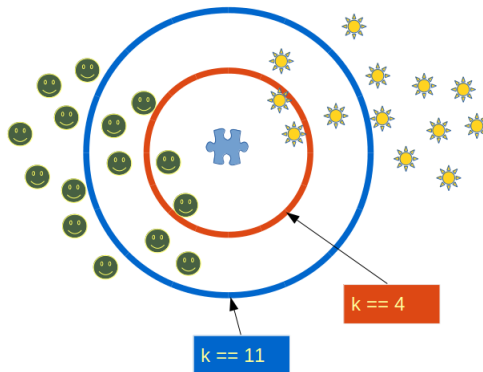
A one-to-one approach was used to classify each observation.



# K Nearest Neighbors

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

🧩 == 😊 or 🧩 == ☀️ ?



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

*DEFINITION :*

Let  $M_1, \dots, M_5$  be our 5 prediction methods (models)

Let  $m_1, \dots, m_5$  be the realized predictions for the 5 methods

Let  $\mathbf{M} = (M_1 = m_1, \dots, M_5 = m_5)$  be the prediction set

$$P(A|\mathbf{M}) \propto P(M_1 = m_1|A) * \dots * P(M_5 = m_5|A) * P(A)$$

$$\vdots$$

$$P(Z|\mathbf{M}) \propto P(M_1 = m_1|Z) * \dots * P(M_5 = m_5|Z) * P(Z)$$

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

Pred M1	Actual Letter		
	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

	<i>B</i>	<i>C</i>	<i>G</i>	<i>K</i>	<i>Q</i>	<i>R</i>
<i>B</i>	629	0	13	2	6	32
<i>C</i>	1	570	38	44	1	0
<i>G</i>	5	43	538	12	46	11
<i>K</i>	7	18	4	574	0	43
<i>Q</i>	11	1	22	0	599	1
<i>R</i>	41	1	7	26	2	603

# Random Forest Results

	<i>H</i>	<i>I</i>	<i>J</i>	<i>K</i>	<i>R</i>
<i>H</i>	671	0	0	19	15
<i>I</i>	0	717	24	0	0
<i>J</i>	1	23	715	1	0
<i>K</i>	9	0	0	701	15
<i>R</i>	3	0	0	9	727

# Support Vector Machines Results

	<i>D</i>	<i>H</i>	<i>I</i>	<i>J</i>
<i>D</i>	780	5	0	0
<i>H</i>	18	669	0	0
<i>I</i>	0	0	727	20
<i>J</i>	2	1	18	717

# Boosting Results

	<i>F</i>	<i>I</i>	<i>J</i>	<i>K</i>	<i>P</i>	<i>R</i>
<i>F</i>	734	3	3	1	11	0
<i>I</i>	6	710	27	0	3	0
<i>J</i>	2	24	706	0	1	0
<i>K</i>	1	0	0	687	0	20
<i>P</i>	16	1	0	0	765	0
<i>R</i>	0	0	0	8	1	715

# K Nearest Neighbors Results

	<i>D</i>	<i>I</i>	<i>H</i>	<i>J</i>	<i>K</i>	<i>F</i>	<i>P</i>
<i>D</i>	774	1	8	0	1	0	0
<i>I</i>	1	725	0	26	0	1	0
<i>H</i>	16	0	637	0	26	0	2
<i>J</i>	0	26	1	707	0	3	0
<i>K</i>	1	0	25	0	669	0	0
<i>F</i>	3	3	0	1	0	716	28
<i>P</i>	2	0	3	0	0	36	749

# Majority Rules Results

	<i>D</i>	<i>F</i>	<i>H</i>	<i>I</i>	<i>J</i>	<i>P</i>
<i>D</i>	790	1	15	1	0	1
<i>F</i>	0	752	0	5	1	17
<i>H</i>	3	1	672	0	1	2
<i>I</i>	0	0	0	724	21	0
<i>J</i>	0	0	0	21	716	0
<i>P</i>	0	8	0	1	0	773

# Bayes Symphony Machines Results

Perfect predictions for A's

	<i>D</i>	<i>F</i>	<i>H</i>	<i>I</i>	<i>J</i>	<i>P</i>
<i>D</i>	790	1	10	0	0	0
<i>F</i>	0	760	0	3	0	16
<i>H</i>	3	1	695	0	0	1
<i>I</i>	0	0	0	729	19	0
<i>J</i>	0	0	0	22	728	0
<i>P</i>	0	8	0	1	0	781



# 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!)

# Distribution of Work

- Problem Statement and Understanding : Zach
- Methods and Model : Joint
- Results : Josh
- Conclusion : Zach
- Cross Validation Code: Josh
- Model Code: Simultaneous