Classification Methods

Logistic Regression
Partitioning … Trees

Classification Problems

Models for a categorical response

- Hate speech
- Supreme Court decisions
- Web ratings: Amazon star ratings, filtering phony reviews

Techniques

- Logistic regression for two, multinomial for several
  - Variable selection (stepwise, lasso)
- Classification trees
  - Boosted trees, random forest
- James text summarizes modern approaches

parametric

|

nonparametric

Where's the text?

Regression with lots and lots of indicators

- Columns of document term matrix
- Presents opportunities, with some evident drawbacks

Simple choice often works well

- Easily interpreted (as easy as any dummy variable)
- Sets a baseline for more complex methods

Combine with other features

- No reason not to use other features if available
- Examples
  - wine data: words from tasting notes + alcohol + vintage
  - real estate: words from listing + square footage
  - medicine: doctor's notes + lab measurements

Review: Logistic Regression

Probability model

Two, mutually exclusive categories

Similar to linear regression in many ways

$P(y_i = 1 | x_i) = E(y_i = 1 | x_i) = \mu_i(\beta_0,\beta_1) = 1/(1+\exp(-\beta_0 - \beta_1 x_i))$

Structural form has important implications

probability goes to 0/1 as |X| gets large

coefficients describe log odds

Maximum likelihood

Estimate parameters to maximize joint probability

$\log P(y_1,y_2,\ldots,y_n | X) = \Sigma_i (1-y_i) \log (1-\mu_i) + y_i \log \mu i$

Independence

Nonlinear least squares (iteratively reweighted least squares)

More than two?

Examples

Not every election is a two-party contest!

Multiple candidates in a primary election

Wine varieties

Think of all the types of red wines that exist.

Multinomial logistic regression (unordered categories)

Multinomial distribution replaces the binomial

$P(y_i = k| x_i) = \mu_i(\beta_0,\beta_1) = \exp(-\beta_{k0} - \beta_{ki}x_i)/(\Sigma_k \exp(-\beta_{k0} - \beta_{ki}x_i))$

Constrained to sum to 1

Reduces to binomial in the case of k=2 categories

Interpretation of coefficients is different in this specification

Model Selection

Which features belong in the logistic regression?

Text presents challenge

Suppose we consider picking columns from the document-term matrix as predictive features

Suppose we consider picking combinations of columns from the document-term matrix

Feature selection

Selection criteria such as AIC, BIC, or stepwise choices

Number of choices overwhelm design of criteria

e.g. AIC designed to pick order of polynomial or autoregression

Assumptions not well suited to the problem (eg “true model”)

Speed becomes limiting factor (recall nonlinear estimation)

Penalized Selection

Problem

Goodness-of-fit statistics like R^2 always go up as add features

Maximum likelihood behaves the same way

Overfitting results

Approach

Add a penalty to the likelihood

Adding a parameter must improve the fit more than the penalty added by increasing model complexity

Question

How much penalty does adding a parameter incur?

Lasso

Penalized likelihood

Choices

L_0	$\max_\beta \text{loglike}(\beta) - \lambda \, \#\{\beta_j \neq 0\}$	AIC, BIC
L_1	$\max_\beta \text{loglike}(\beta) - \lambda \, \Sigma \, \lvert\beta_j\rvert$	
L_2	$\max_\beta \text{loglike}(\beta) - \lambda \, \Sigma \, \beta_j^2$	Ridge regr

λ controls the amount of the penalty

Lasso = L_1 penalty

Advantages

Fast computing because objective function is convex

Criterion sets many $\beta_j = 0$, unlike ridge penalty

Penalty Parameter

Choice of tuning parameter λ

- Really big: model is parsimonious
- Really small: model has many features

Bias-Variance tradeoff

- Big models have little bias, but high variance
- Small models reverse this balance

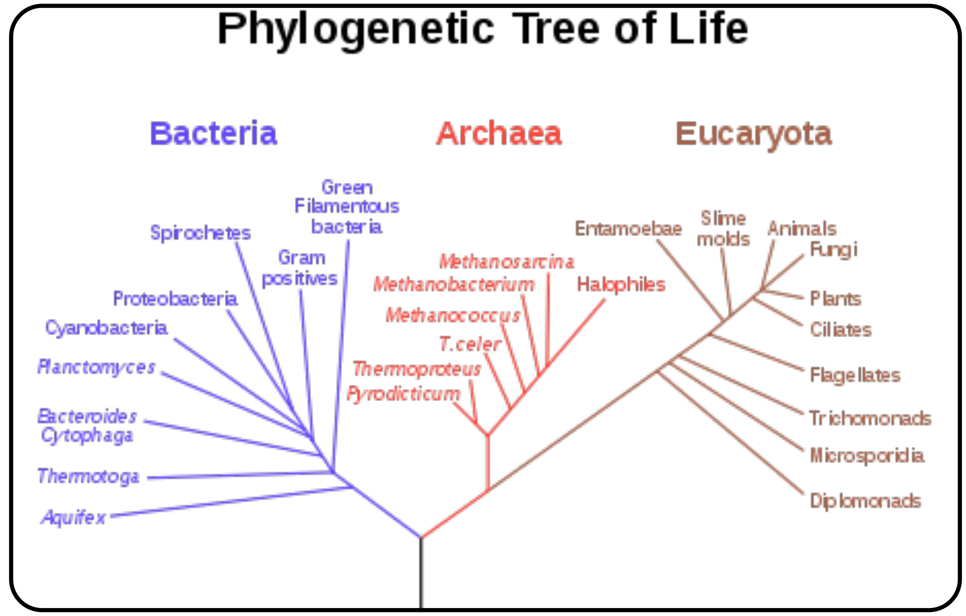
Choice uses cross validation

- Ten-fold cross-validation of the training data
- Fit model to 9/10, predict the other 1/10. Repeat
- Pick λ that minimizes the error

Partitioning Models: Trees

Familiar metaphor

- Biology
- Medical diagnosis
- Org chart



Structure at-a-glance

Properties

- Recursive, partitioning items into unique leaf
- Increasing specialization

How to grow a tree from data?

- What rules identify the splitting variables, split points?

Classical Example

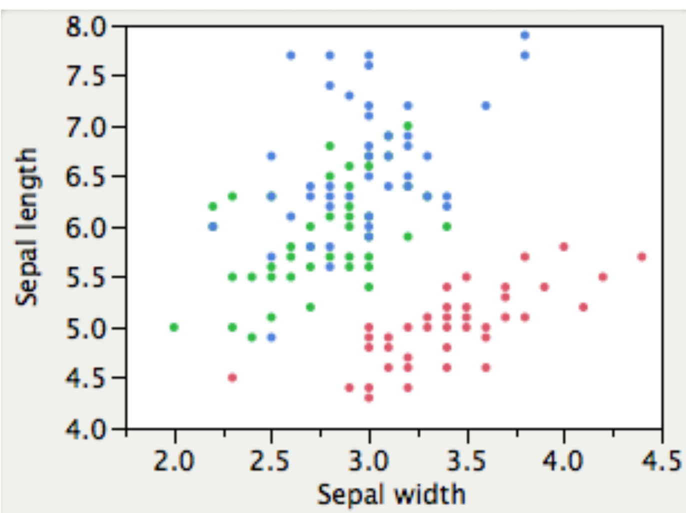
Classification tree

Fisher's iris data

Classification tree: categorical response

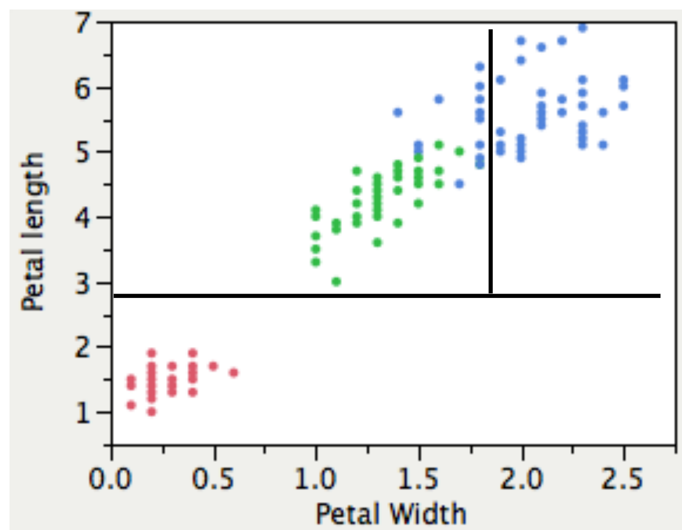
50 flowers from 3 species of iris

four variables: length and width of sepal and petal



Splits are parallel to plot axes

Splitting rules are not unique



Stop?

CART™

Classification and regression trees

A sequence of divisions of cases

Goal is to obtain homogeneous subsets

Predict new observations based on “vote” of leaf

Classification tree

Categorical response (e.g. good/bad/indifferent)

Goal: Cases in leaf belong to one category

Regression tree

Numerical response (e.g. profitability)

Cases in leaf have similar value of response

Familiar likelihood objective

Choose leaves to maximize likelihood

Simple Foundation

Bins, lots of bins

Allow variables (characteristics) to define a large "cube" with dimensions given by

Age x Employment x Residential

Insert each observation into a bin

Score for bin is average of observations in bin

Trade-offs

bias
vs
variance

Don't have to pick additive form, transformations

Some bins may be nearly empty, sparse

Issues remain

Which characteristics? Which attributes?

Goodness of Fit

Two general approaches

Classification error

- Confusion matrix: Count number wrong
  - “Millions” of summary stats: sensitivity, specificity, recall, precision, f1
- What does it mean to be wrong?
- ROC curve and AUC

Proper scoring rules

- Squared error
- Likelihoods

Confusion Matrix

		claim	
		neg	pos
actual	neg	n_{11}	n_{12}
	pos	n_{21}	n_{22}

Confusion matrix

Common summary table

Misclassification rate

Sensitivity & specificity

Sensitivity = P(say positive | positive) = Recall

Specificity = P(say negative | negative)

Precision = P(positive | say positive)

F_1 = 2 (precision x recall)/(precision+recall)   harmonic mean

Classification error rate

Common, but ‘coarse’

What threshold would you use to classify?

ROC Curves

ROC Curve

True positive (sensitivity) vs false positive (1-specificity)

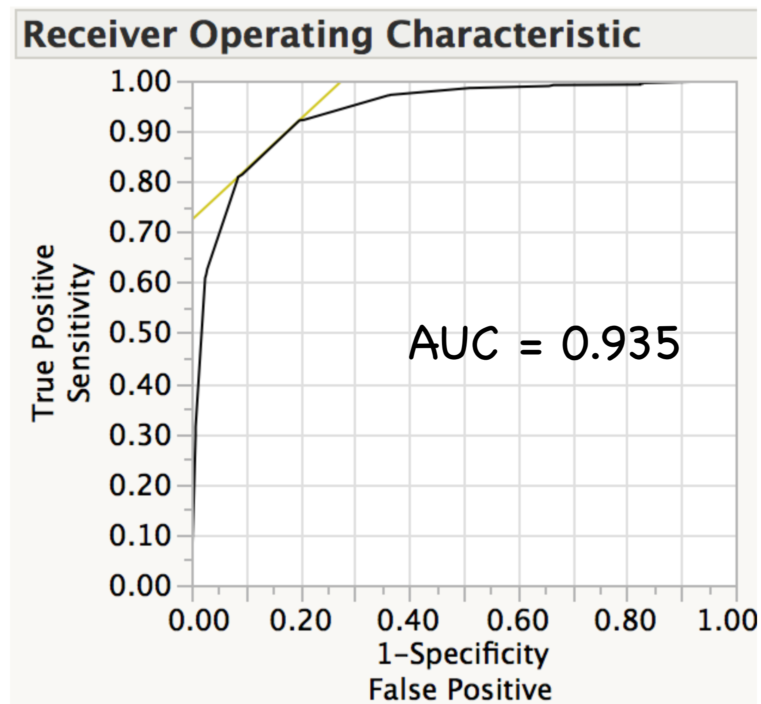
Equivalent to Gini index

Only order matters, not the calibration

AUC

Area under ROC curve

Interpret as probability
fit correctly orders pair



Points of interest?

Care about whole curve?

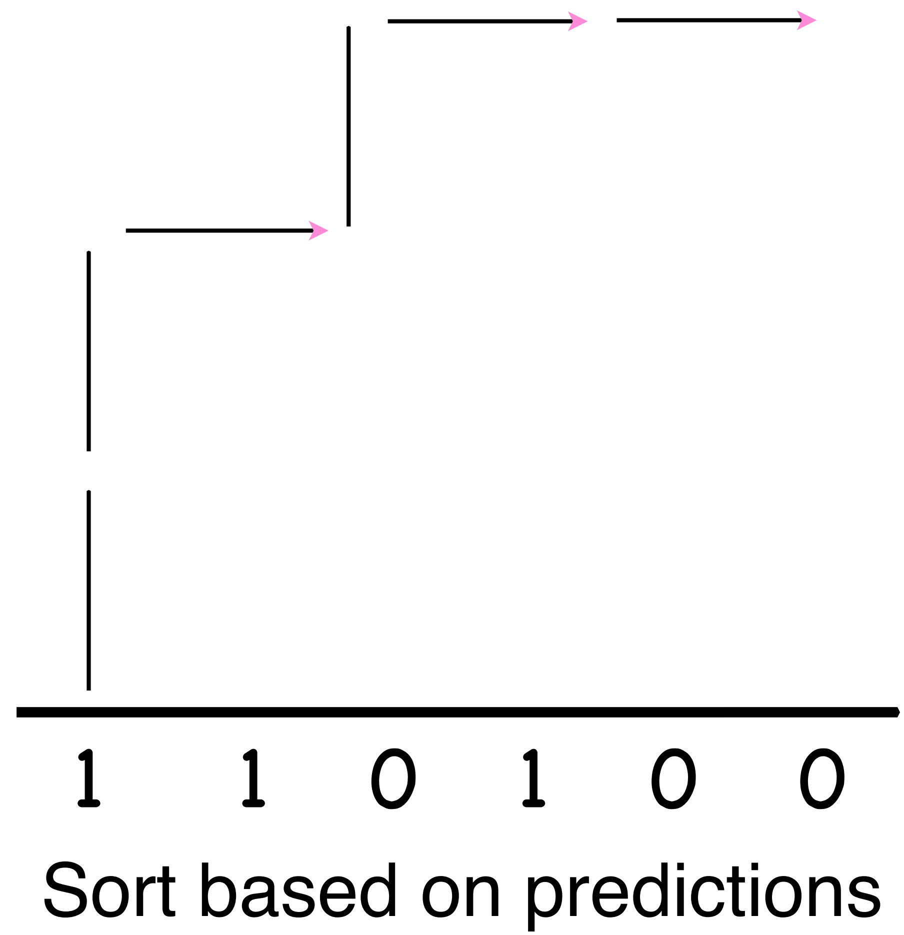
Economics of derivative

Drawing the ROC

Order cases by probabilities

Move up
if positive case

Move right
if negative case



Deviance

Twice the log of the likelihood ratio statistic

Least squares regression. Assume $y_i \sim N(0,\sigma^2)$

Null model

$-2 \text{ loglike}(M_0) = \Sigma(y_i)^2/\sigma^2 \sim$ chi-square n df $= \chi^2_n$

Regression with k estimated coefficients

$-2 \text{ loglike}(M_k) = \Sigma(y_i - \hat{y}_i)^2/\sigma^2 \sim \chi^2_{n-k}$

assuming variables have true coefficient $\beta_k=0$

Change in log-likelihood when add nothing useful:

$-2(\text{loglike}(M_0) - \text{loglike}(M_k)) \sim \chi^2_k$

Deviance

-2 (loglike(base model) - loglike(fitted model)) $\sim \chi^2_{\text{estimated parms}}$

Validation

Necessary when comparing complex models

- Easy to overfit complex models
  - Model might have more potential features than observations
  - Eg: Occurrence of which pairs of words indicate how Justice will decide?
- Keep changing model until it fits the observed data all too well

Validation?

- Assess goodness of fit on a test set, not training data
- How many?
  - Depends on task: are models similar

Caution: Test set gives optimistic assessment

- Population drift

Improving Trees

Bias-variance trade-off

Analogous to choice of smoothing parameter

Trees capture nuanced structure, but (low bias)

Trees have highly irregular structure (high var)

Model averaging

Rather than fit one model, fit several and combine results

- Classifier: majority vote
- Regression: average predictions

Approaches

Boosting	“stumps” or small trees are so-called weak learners
Bagging	bootstrap resampling method

Boosting

General method for improving any simple model

Build sequence of predictive models...

- Start with initial predictive model
- Compute residuals from current fit
- Build model for residuals
- Repeat
- Combine estimates from sequence of models

Use simpler model at each step

- Small tree (stump or bush)
- Next response = (current response) - (learning rate) x fit

Weaknesses

- Loss of interpretability, at what gain?

Adaboost

reweighting cases

Boosting Trees

Pick depth of tree (stumps), learning rate

Use cross-validation to pick B

Analogous to picking λ for logistic models

Algorithm 8.2 *Boosting for Regression Trees*

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
   (a) Fit a tree $\hat{f}^b$ with d splits ($d+1$ terminal nodes) to the training data (X, r).
   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x). \tag{8.10}$$
   (c) Update the residuals,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \tag{8.11}$$
3. Output the boosted model,
$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \tag{8.12}$$

James
Ch 8

Classification Examples

wine_classify.R

Plan

Predicting wine color

- Two-category response
- Easy for both logistic regression and tree

Predicting the type of wine

- Four-category problem
- More challenging
  - Harder to distinguish from choices of words
  - Fewer observations to build a model

Judging models

- Common test sample hidden from each method

Predicting Wine Color

Red or white?

Combine columns from DTM with other data

- Indicators or counts
  - Do we care about how often a word was used, or just its presence?
- Lengths and proportions
  - Is the count most relevant, or the relative frequency
- Choice of predictors is up to you!

Note: missing data in the other features!

- 10% missing vintage or price, 2.5% missing alcohol — no pun intended
- Use same approach as in linear regression

Logistic Model

Exclude test sample from all models

Set aside 10,000 …

Why: Test accuracy, and this will make modeling harder

Start with the classic variables

price, alcohol, vintage, missing indicators, and lengths

Interpretation?

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.152e+02  1.904e+01  11.303  < 2e-16
alcohol      7.590e-01  3.326e-02  22.822  < 2e-16
vintage     -1.133e-01  9.501e-03 -11.920  < 2e-16
price        9.091e-04  2.364e-03   0.385  0.70055
lengths      5.109e-02  2.982e-03  17.132  < 2e-16
Miss.alcohol 3.777e+00  5.147e-01   7.337 2.18e-13
Miss.vintage 3.550e-01  1.261e-01   2.816  0.00486
Miss.price  -6.591e-01  1.036e-01  -6.359 2.02e-10

    Null deviance: 9890.1  on 7335  degrees of freedom
Residual deviance: 8253.2  on 7328  degrees of freedom
AIC: 8269.2
```

price isn't but missing is

Logistic with Words

Which words

Start with simply using proportions of 20 most common words

Common words useful … proxies for length?

(Intercept)	0.6879	0.3192	2.155	0.031148
w_comma_	6.3219	0.9612	6.577	4.80e-11
w_and	2.5392	1.3865	1.831	0.067043
w_period_	-6.3340	1.7254	-3.671	0.000242
w_dash_	-5.6080	1.4712	-3.812	0.000138
w_with	8.2931	1.7676	4.692	2.71e-06
w_aromas	-21.3342	3.4611	-6.164	7.09e-10
w_medium	-1.9850	3.0651	-0.648	0.517240
w_finish	-20.8601	2.4032	-8.680	< 2e-16

…

```
    Null deviance: 9890.1  on 7335  degrees of freedom
Residual deviance: 6507.9  on 7315  degrees of freedom
AIC: 6549.9
```

much less residual deviance

Logistic with Words

Which words

Add length to the mixture

Effects still strong for common words, conditional on length

Interpret?

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.606257   0.469227  -3.423 0.000619
lengths       0.032010   0.004803   6.664 2.66e-11
w_comma_      6.346087   0.967310   6.561 5.36e-11
w_and         4.040886   1.407849   2.870 0.004101
w_period_    -1.890119   1.852974  -1.020 0.307706
w_dash_      -6.088978   1.482897  -4.106 4.02e-05
w_with        8.561713   1.769272   4.839 1.30e-06
w_aromas    -13.798451   3.637871  -3.793 0.000149
w_medium      0.979913   3.115882   0.314 0.753149
w_finish    -15.881983   2.517048  -6.310 2.79e-10
```

...

```
    Null deviance: 9890.1  on 7335  degrees of freedom
Residual deviance: 6462.4  on 7314  degrees of freedom
AIC: 6506.4
```

Logistic with Both

Combine two prior models

Observed quantitative features

Word relative frequencies + length

	Estimate	Std. Error	z value
(Intercept)	484.201035	40.188341	12.048
alcohol	0.765631	0.042116	18.179
vintage	-0.247354	0.020033	-12.348
price	-0.003278	0.002830	-1.158
lengths	0.028724	0.005432	5.288
Miss.alcohol	3.627510	0.532791	6.809
Miss.vintage	-0.524793	0.171330	-3.063
Miss.price	-0.608658	0.132712	-4.586
w_comma_	6.213209	1.056403	5.881
w_and	4.533635	1.541723	2.941
w_period_	0.037003	2.073927	0.018
w_dash_	-3.183346	1.631417	-1.951
w_with	10.122619	1.934195	5.234
w_aromas	-20.266659	4.326438	-4.684

```
    Null deviance: 9890.1  on 7335  degrees of freedom
Residual deviance: 5596.4  on 7308  degrees of freedom
AIC: 5652.4
```

add more?

Logistic with More Words

Extend prior model

Observed quantitative features

40 Word relative frequencies + length

hints of collinearity

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	399.593838	55.866871	7.153	8.51e-13
alcohol	0.675543	0.053915	12.530	< 2e-16
vintage	-0.203908	0.027800	-7.335	2.22e-13
price	-0.001677	0.004058	-0.413	0.679385
lengths	0.014362	0.007167	2.004	0.045065
Miss.alcohol	4.146267	0.678135	6.114	9.70e-10
Miss.vintage	-0.357215	0.214794	-1.663	0.096300
Miss.price	-0.485995	0.182512	-2.663	0.007749
w_comma_	3.654130	1.504159	2.429	0.015126
w_and	0.700532	2.238983	0.313	0.754372
w_period_	-0.447489	2.940249	-0.152	0.879034
w_dash_	0.754281	2.756884	0.274	0.784393
w_with	6.014786	3.127141	1.923	0.054428
w_aromas	-16.173814	5.703468	-2.836	0.004571

```
    Null deviance: 9890.1  on 7335  degrees of freedom
Residual deviance: 3366.6  on 7288  degrees of freedom
AIC: 3462.6
```

much better fit!

add more?

Wharton
Department of Statistics

Test Model

Predict color of wines held back in the test sample

```
                     | pred > 0.5
Data[test, "color"]  |     FALSE |      TRUE | Row Total |
---------------------|-----------|-----------|-----------|
                   0 |      3544 |       424 |      3968 |
      White          |     0.893 |     0.107 |     0.397 |
                     |     0.878 |     0.071 |           |
---------------------|-----------|-----------|-----------|
                   1 |       492 |      5540 |      6032 |
      Red            |     0.082 |     0.918 |     0.603 |
                     |     0.122 |     0.929 |           |
---------------------|-----------|-----------|-----------|
        Column Total |      4036 |      5964 |     10000 |
                     |     0.404 |     0.596 |           |
---------------------|-----------|-----------|-----------|
```

sensitivity	0.918
specificity	0.893
precision	0.929
missclass	0.092

```
  Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|-------------------------|
```

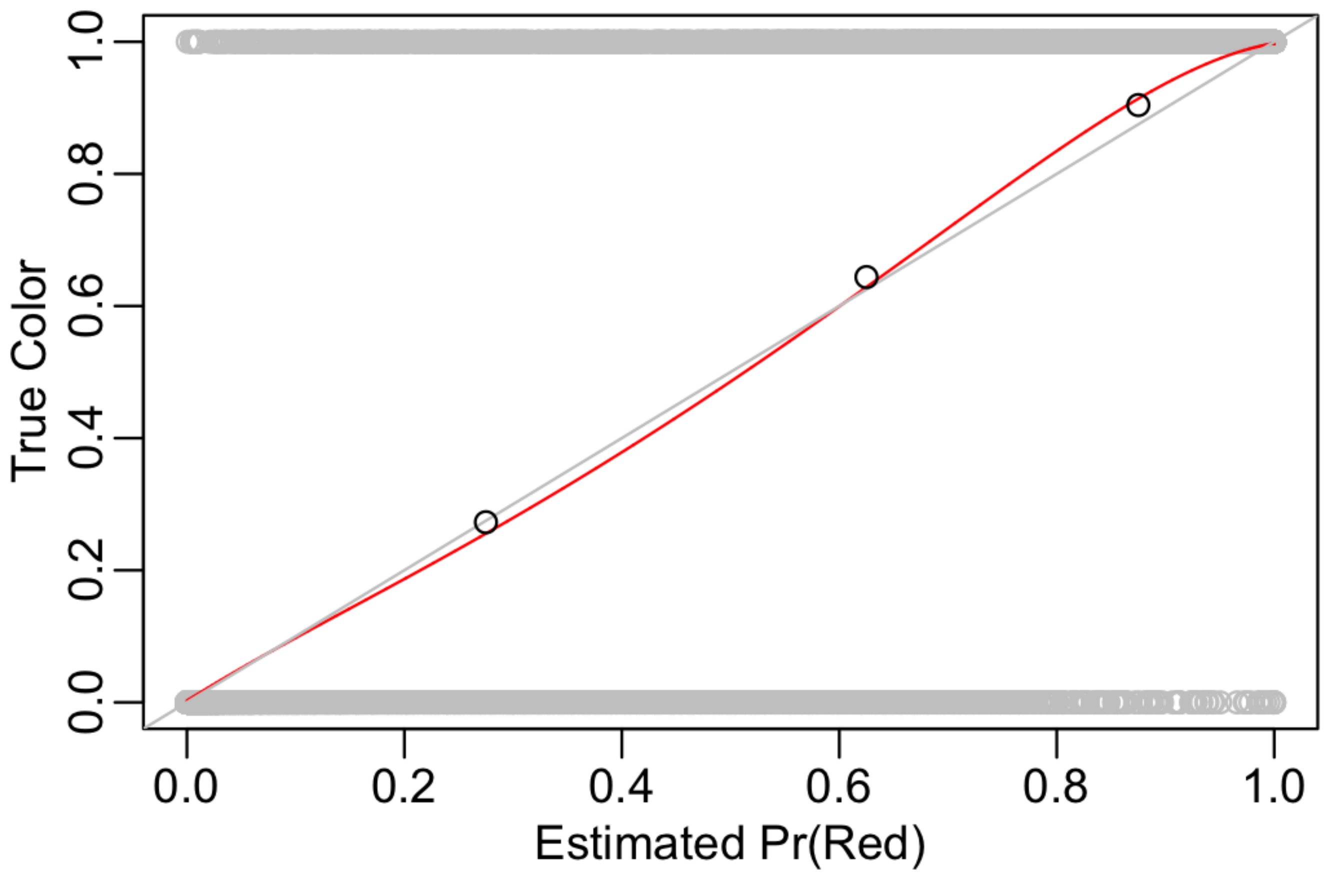
precision= # Red/# Claim Red
recall = sensitivity = #Claim Red/# Red

Calibration

Do predicted probabilities indicate actual probability?

Hosmer-Lemeshow test

Plot adds high-degree polynomial (or loess smooth curve)



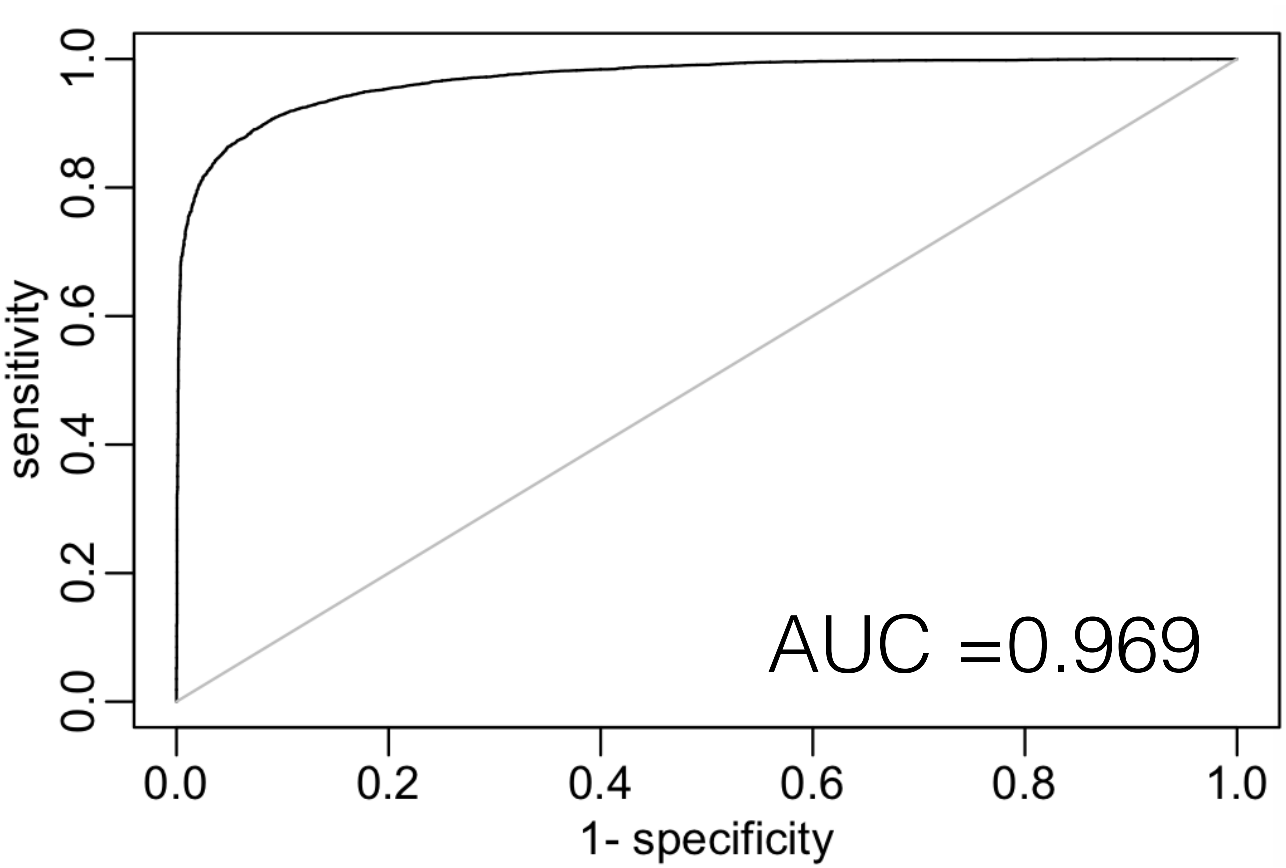
Not a problem if threshold at 0.5

ROC Curve

Plot sensitivity on 1-specificity

Parametric curve as vary the classification threshold

true positive



false positive

Variable Selection

Which words

- Twenty words was good, forty was better
- Keep going… we have thousands

Try feature selection

- Stepwise logistic regression is slow
- Lasso in R offers fast alternative
  - glmnet package is very efficient
- Dimension of the DTM is a challenge these tools
  - Estimation data has 7336 cases with 2659 word columns

Baseline

- Models already achieve in-sample residual deviance 3367

Lasso Selection

Start with set of features from prior logistic regression

Basic variables (alcohol, price, etc)

Proportions of top 40 words

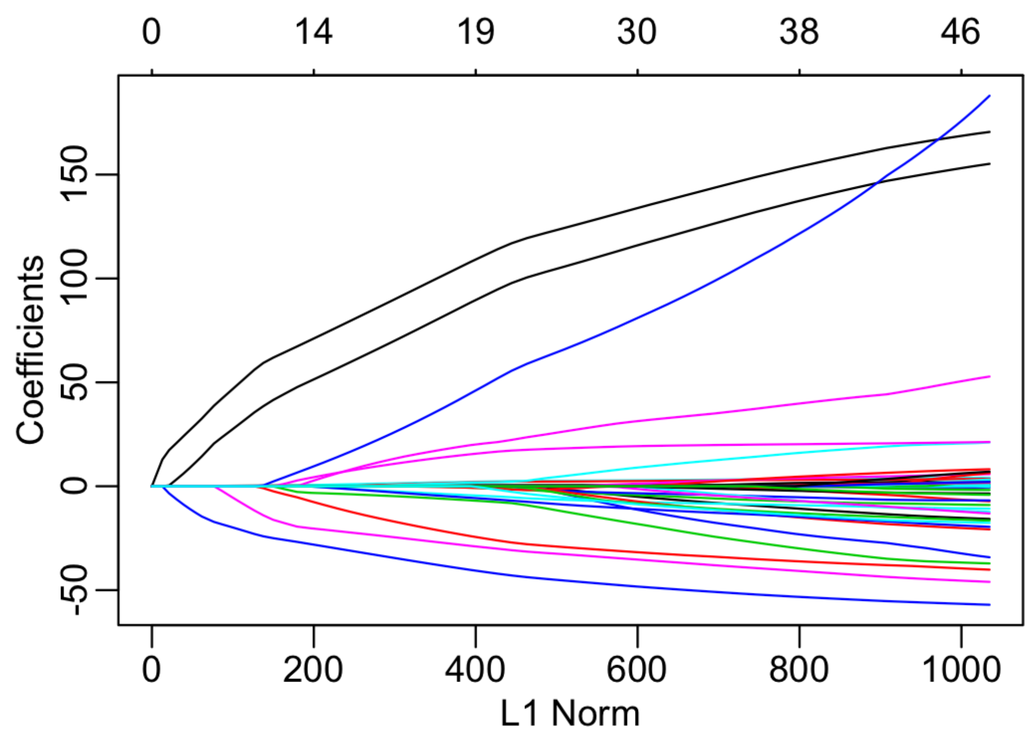
Fishbone plot

Coefs as reduce penalty λ

Trace each as λ –> 0

Far right is logistic model

```
48 x 1 sparse Matrix of class
                          1
(Intercept)   399.885896578
alcohol         0.675597222
vintage        -0.204053180
price          -0.001677798
lengths         0.014367030
Miss.alcohol    4.145649206
Miss.vintage   -0.357713954
Miss.price     -0.486269834
w_comma_        3.650720327
```



analogous to ridge trace

How many to use

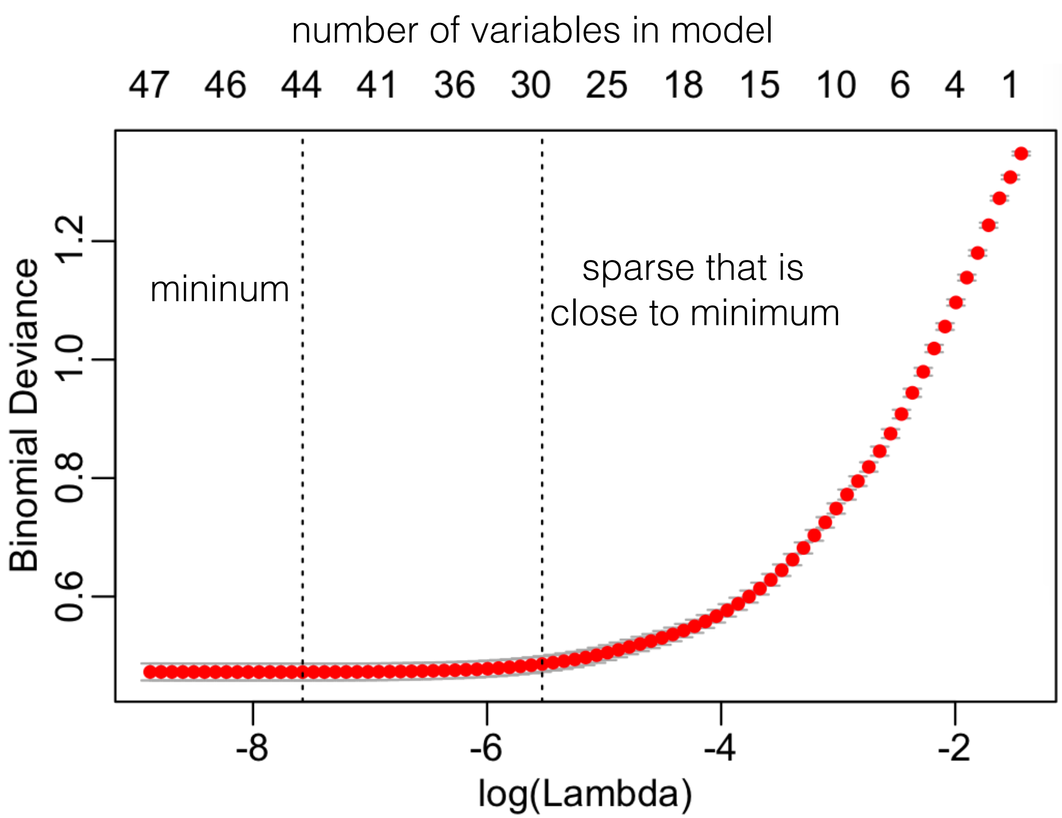
Pick value of λ using cross validation

10-fold cross-validation

10 splits of training data (not using held back test sample)

distinguish training from tuning from testing

Best model is not very sparse



Again find the "long tail" of signal in text

Wharton
Department of Statistics

Performance

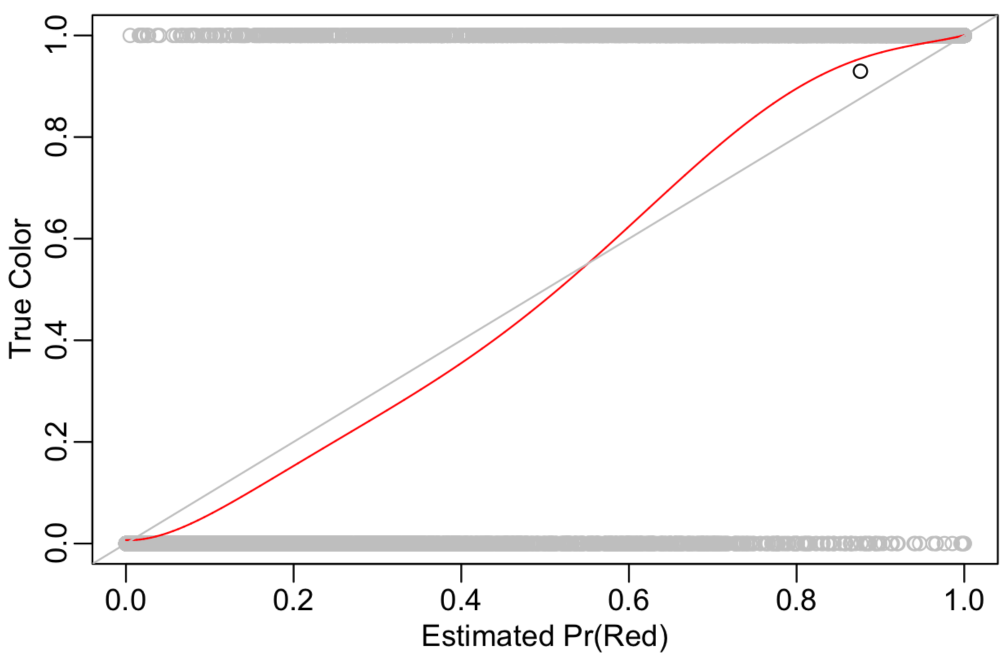
Use sparse model within 1 SE of minimum

17 coefficients are zeroed out, leaving 31 estimates

Similar to prior logistic regression, but with 17 fewer estimates

Not so well calibrated away from 0.5, our threshold

Confusion matrix provides matching results



	LR	Lasso
sensitivity	0.918	0.915
specificity	0.893	0.891
precision	0.929	0.928
missclass	0.092	0.094

Use More Words!

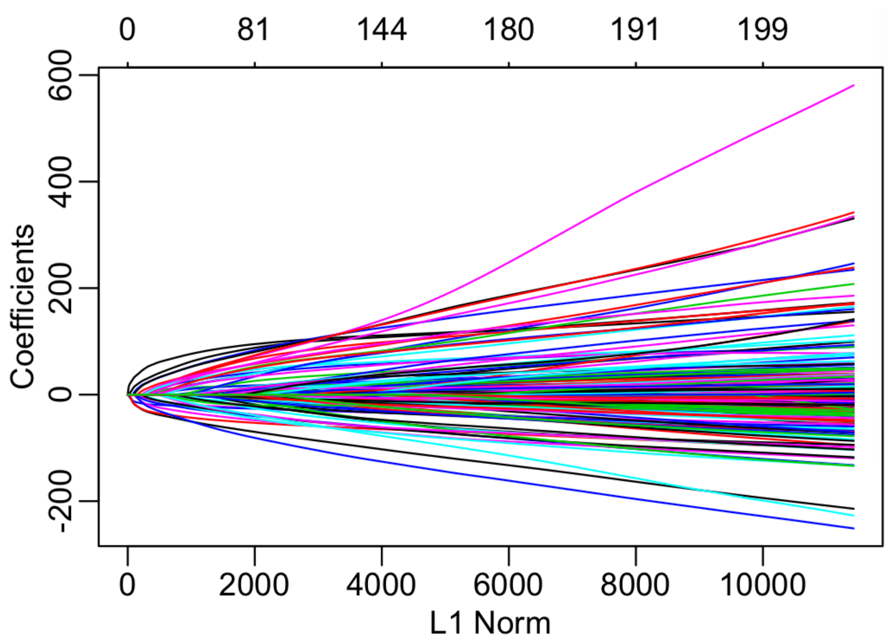
Cast a bigger net

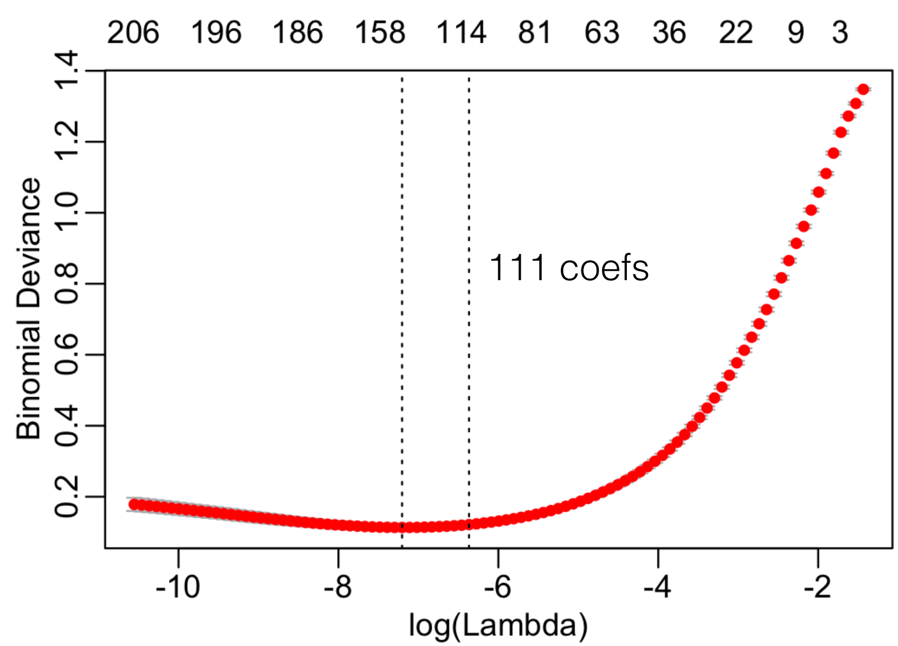
Try to use Lasso to pick from wider collection of words

Speed decreases

Initial fitting is fast, but picking λ by 10-fold CV slows the process

would like a progress indicator!

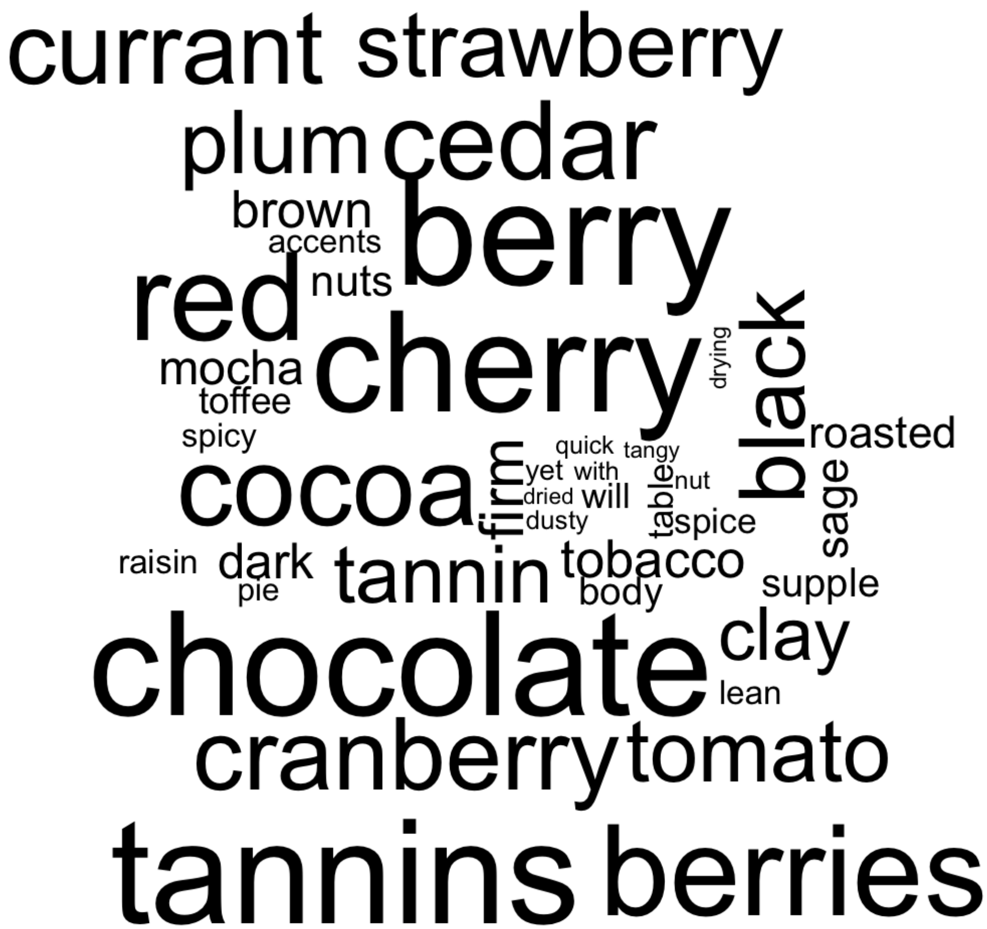




Wharton
Department of Statistics

What are the coefficients?

Use a word cloud, weighted by the estimates…



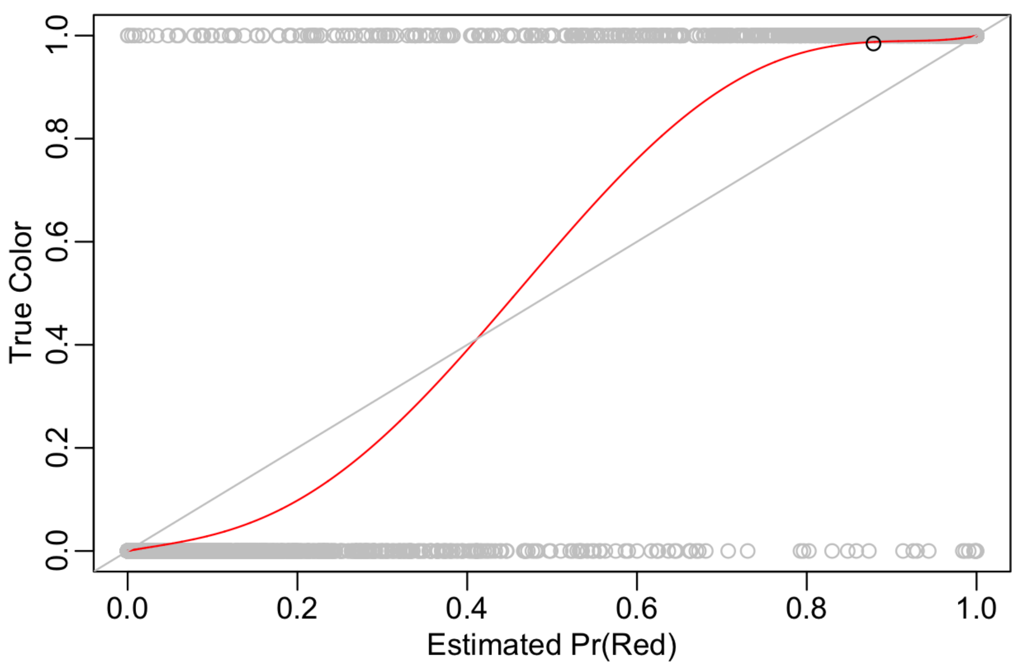
Nice to see the word ‘red’!

How well did it work?

Comparison in the test set…

Calibration getting far off target away from 0.5

Logistic model no longer working



	LR	Lasso	200
sens	0.918	0.915	0.982
spec	0.893	0.891	0.987
prec	0.929	0.928	0.991
miss	0.092	0.094	0.016

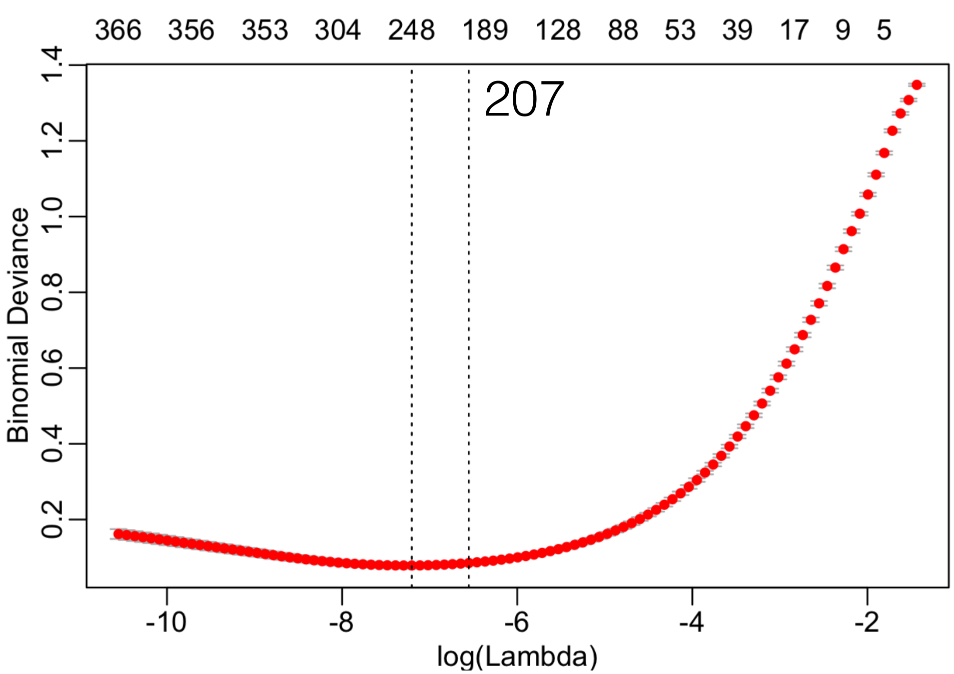
Quite an improvement

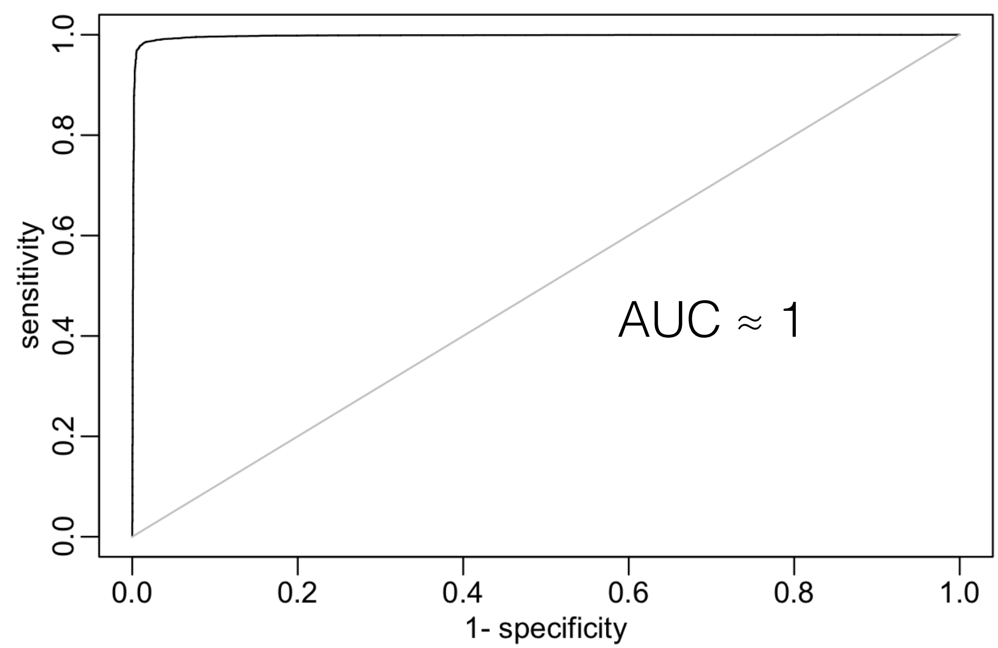
More?

Try with 500 words in model…

Fitting remains fast, with CV slowing the process… but not that much.

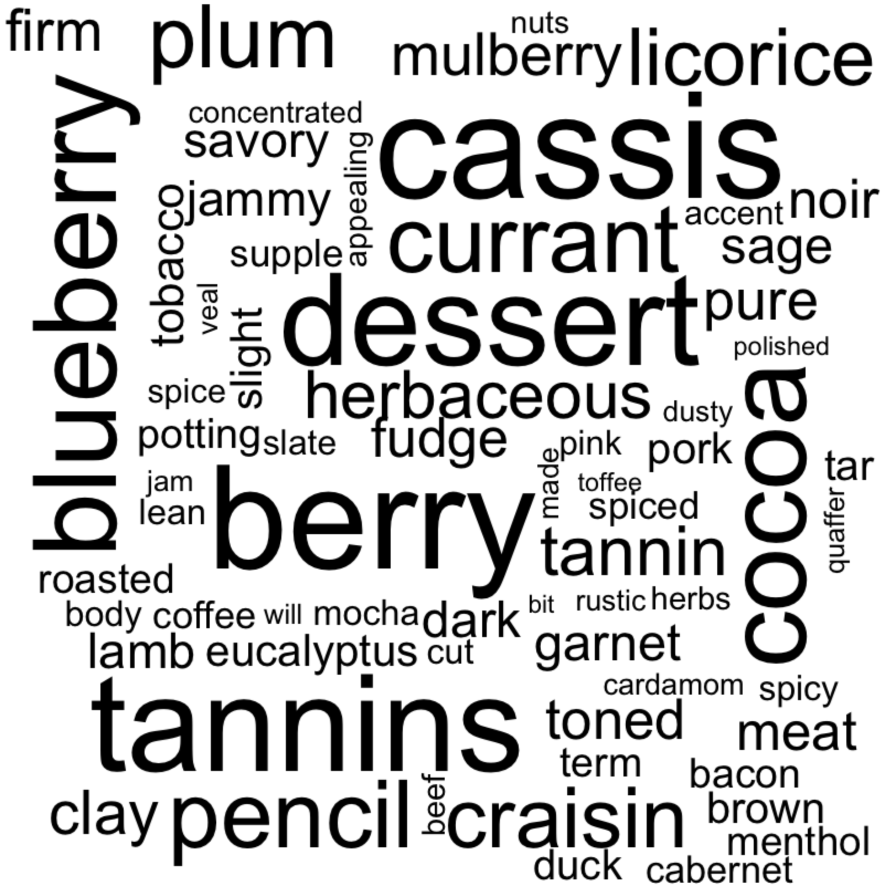
Similar confusion matrix





But different words…

- Similar fit, but many different words
  - Collinearity becoming an issue

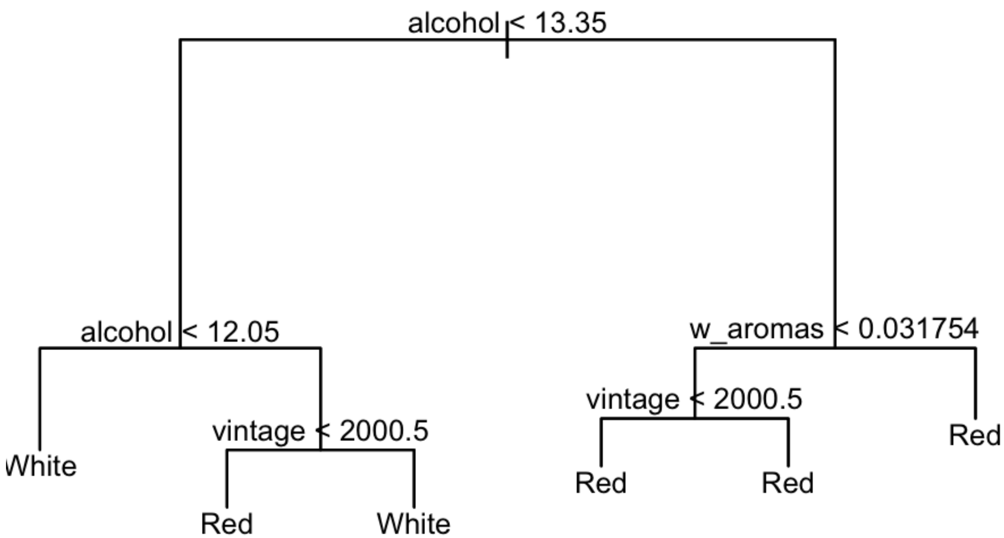


Change Direction: Trees

Try a different type of model: a classification tree

Example with a few words

Classify using majority vote



deviance in node

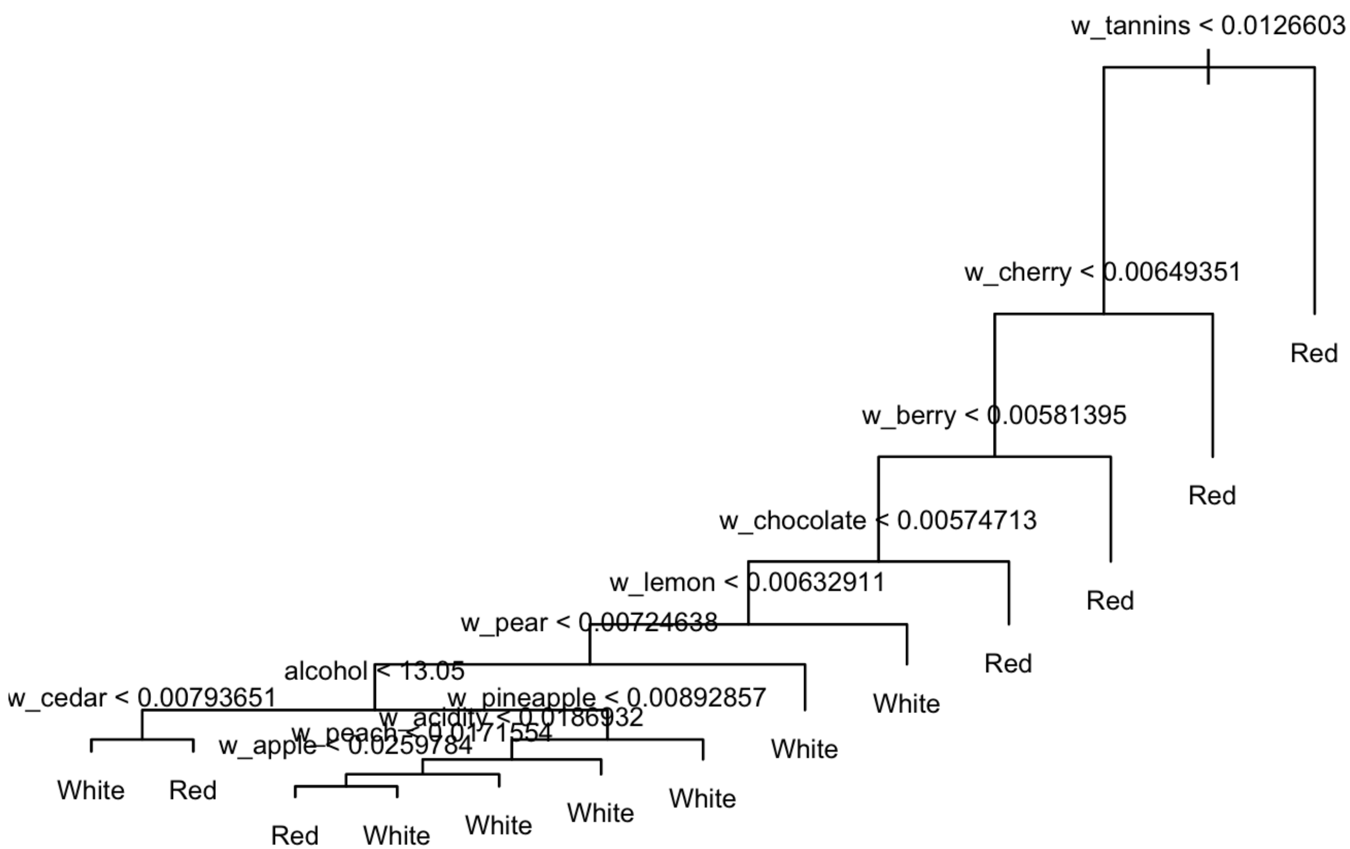
```
1) root 7336 9890.0 Red ( 0.59733 0.40267 )
  2) alcohol < 13.35 2883 3890.0 White ( 0.40409 0.59591 )
    4) alcohol < 12.05 779  730.7 White ( 0.17843 0.82157 ) *
    5) alcohol > 12.05 2104 2915.0 White ( 0.48764 0.51236 )
     10) vintage < 2000.5 289  289.8 Red ( 0.79931 0.20069 ) *
     11) vintage > 2000.5 1815 2488.0 White ( 0.43802 0.56198 ) *
  3) alcohol > 13.35 4453 5260.0 Red ( 0.72243 0.27757 )
    6) w_aromas < 0.031754 3226 3420.0 Red ( 0.77743 0.22257 )
     12) vintage < 2000.5 501  227.1 Red ( 0.94012 0.05988 ) *
     13) vintage > 2000.5 2725 3079.0 Red ( 0.74752 0.25248 ) *
    7) w_aromas > 0.031754 1227 1671.0 Red ( 0.57783 0.42217 ) *

Number of terminal nodes:  6
Residual mean deviance:  1.158 = 8486 / 7330
Misclassification error rate: 0.3037 = 2228 / 7336
```

Bigger Tree

Use 1000 words

- Fitting a tree is surprisingly fast
- Shape conveys the value of certain words



Some Details

Inspect the terminal nodes

```
Number of terminal nodes:  13
Residual mean deviance:  0.3449 = 2526 / 7323
Misclassification error rate: 0.05889 = 432 / 7336
```

```
1) root 7336 9890.000 Red ( 0.597328 0.402672 )
  2) w_tannins < 0.0126603 5263 7226.000 White ( 0.442333 0.557667 )
    4) w_cherry < 0.00649351 4444 5707.000 White ( 0.341584 0.658416 )
      8) w_berry < 0.00581395 3994 4642.000 White ( 0.267902 0.732098 )
       16) w_chocolate < 0.00574713 3726 3915.000 White ( 0.218733 0.781267 )
         32) w_lemon < 0.00632911 2932 3456.000 White ( 0.276262 0.723738 )
           64) w_pear < 0.00724638 2241 2924.000 White ( 0.358322 0.641678 )
            128) alcohol < 13.05 1043  971.900 White ( 0.176414 0.823586 )
              256) w_cedar < 0.00793651 1007  844.200 White ( 0.147964 0.852036 ) *
              257) w_cedar > 0.00793651 36    9.139 Red ( 0.972222 0.027778 ) *
            129) alcohol > 13.05 1198 1659.000 Red ( 0.516694 0.483306 )
              258) w_pineapple < 0.00892857 1073 1462.000 Red ( 0.576887 0.423113 )
                516) w_acidity < 0.0186932 824 1037.000 Red ( 0.677184 0.322816 )
                 1032) w_peach < 0.0171554 766  899.800 Red ( 0.725849 0.274151 )
                   2064) w_apple < 0.0259784 668  680.600 Red ( 0.793413 0.206587 ) *
                   2065) w_apple > 0.0259784 98  113.400 White ( 0.265306 0.734694 ) *
                 1033) w_peach > 0.0171554 58   17.400 White ( 0.034483 0.965517 ) *
                517) w_acidity > 0.0186932 249  277.300 White ( 0.244980 0.755020 ) *
              259) w_pineapple > 0.00892857 125    0.000 White ( 0.000000 1.000000 ) *
           65) w_pear > 0.00724638 691   78.220 White ( 0.010130 0.989870 ) *
         33) w_lemon > 0.00632911 794   60.640 White ( 0.006297 0.993703 ) *
       17) w_chocolate > 0.00574713 268  104.000 Red ( 0.951493 0.048507 ) *
      9) w_berry > 0.00581395 450   25.660 Red ( 0.995556 0.004444 ) *
    5) w_cherry > 0.00649351 819   99.100 Red ( 0.989011 0.010989 ) *
  3) w_tannins > 0.0126603 2073  216.100 Red ( 0.990835 0.009165 ) *
```

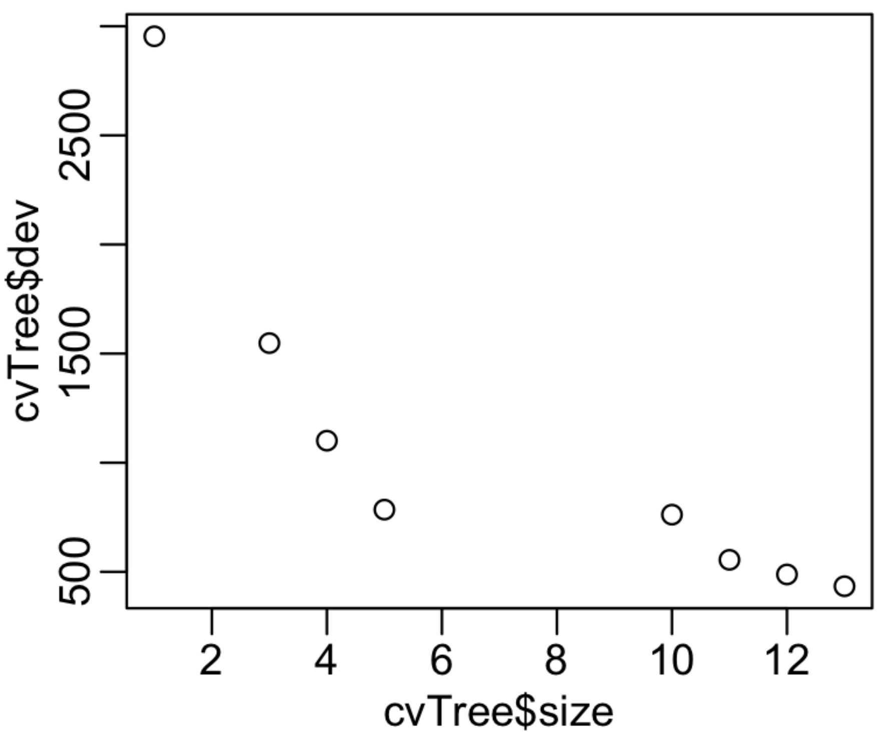
Better Tree-based Classifier

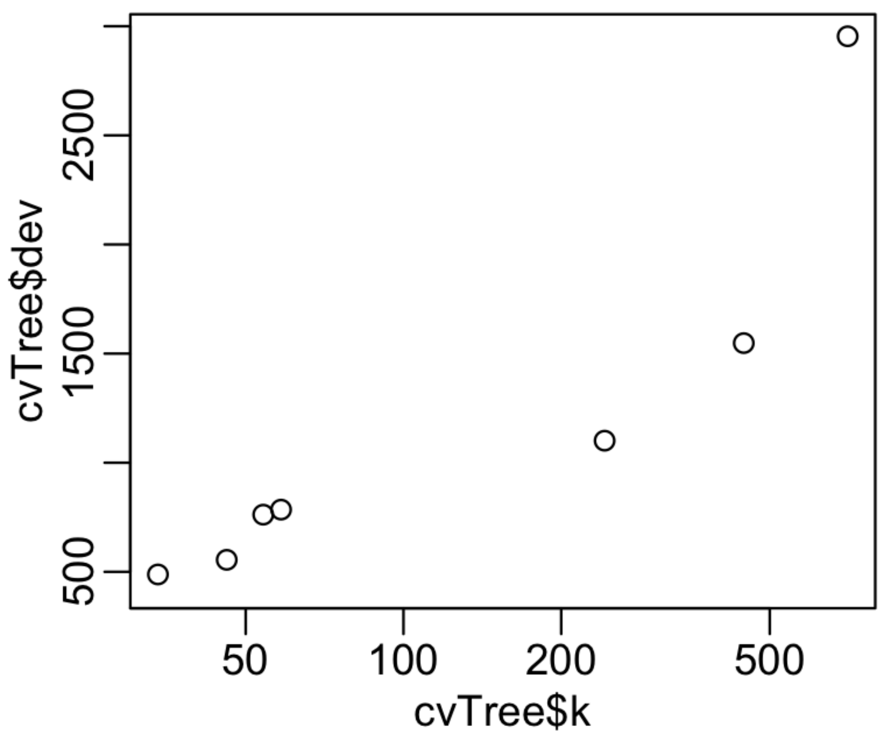
Prune tree

Use cross-validation to remove nodes

Smaller tree often classifies better, avoiding overfitting

In this case, retains tree with 13 terminal nodes



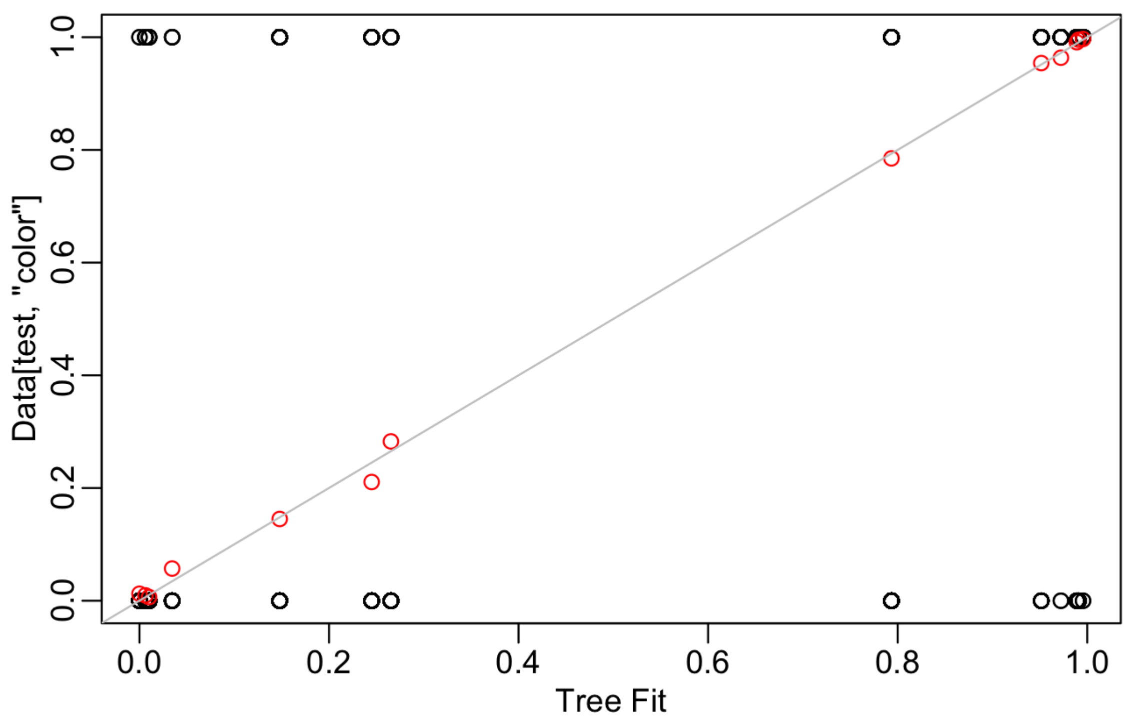


Boosted Trees

Smooth out the discontinuity of tree fits

Number of distinct predictions = number of terminal nodes

Averaging over many small trees smooths predictions



Boosted Results

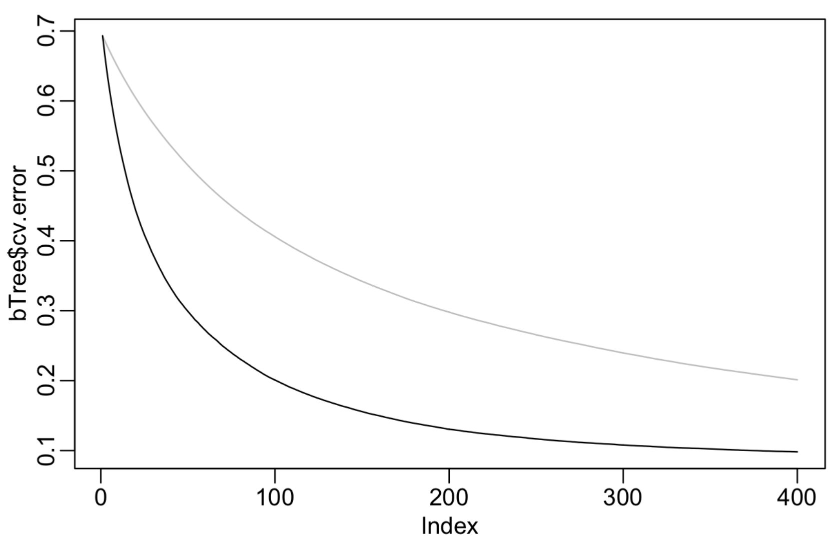
Using 400 words

Code is not so fast again as was the case with

Fitting process incorporates CV to control boosting process

- That's where code can die if a word appears in test, but not training
- Seems to happen in 'bernoulli' mode, but not for multinomial

Fit as learning progresses

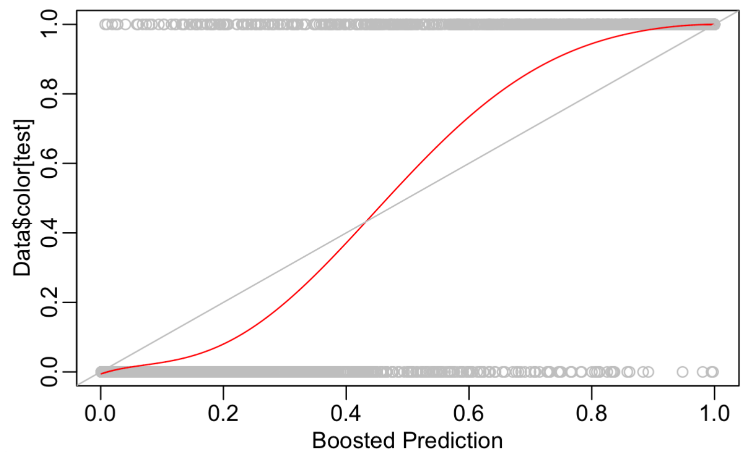


slower is better,
but
slower is slower

Boosted Performance

Using 400 words…

Predictions range over [0,1]



Much more competitive, but not up to level of the regression!

	LR	Lasso	200	BT
s	0.918	0.915	0.982	0.969
spec	0.893	0.891	0.987	0.974
prec	0.929	0.928	0.991	0.983
miss	0.092	0.094	0.016	0.029

Wharton
Department of Statistics

Predicting Variety

Predicting wine variety

- Four-category response: cabernet, merlot, pinot, zinfandel
- Smaller sample size
- Much more similar in nature of descriptions

Multinomial regression

- Generalization of logistic regression to more than two groups
- Trees generalize directly… just more labels

Comparing models

- Common test sample hidden from each method

Varieties

Possible choices

```
Chardonnay Cabernet Sauvignon             Merlot         Pinot Noir    Sauvignon Blanc
      2215               1873               1250               1087                883
 Zinfandel           Riesling              Syrah
       696                689                590
```

Choose top four categories of reds, 4,906 tasting notes

Set aside validation cases, 250 for each variety

Limited by number of Zinfandels

Build initial model using numerical features

Baseline for value of adding text

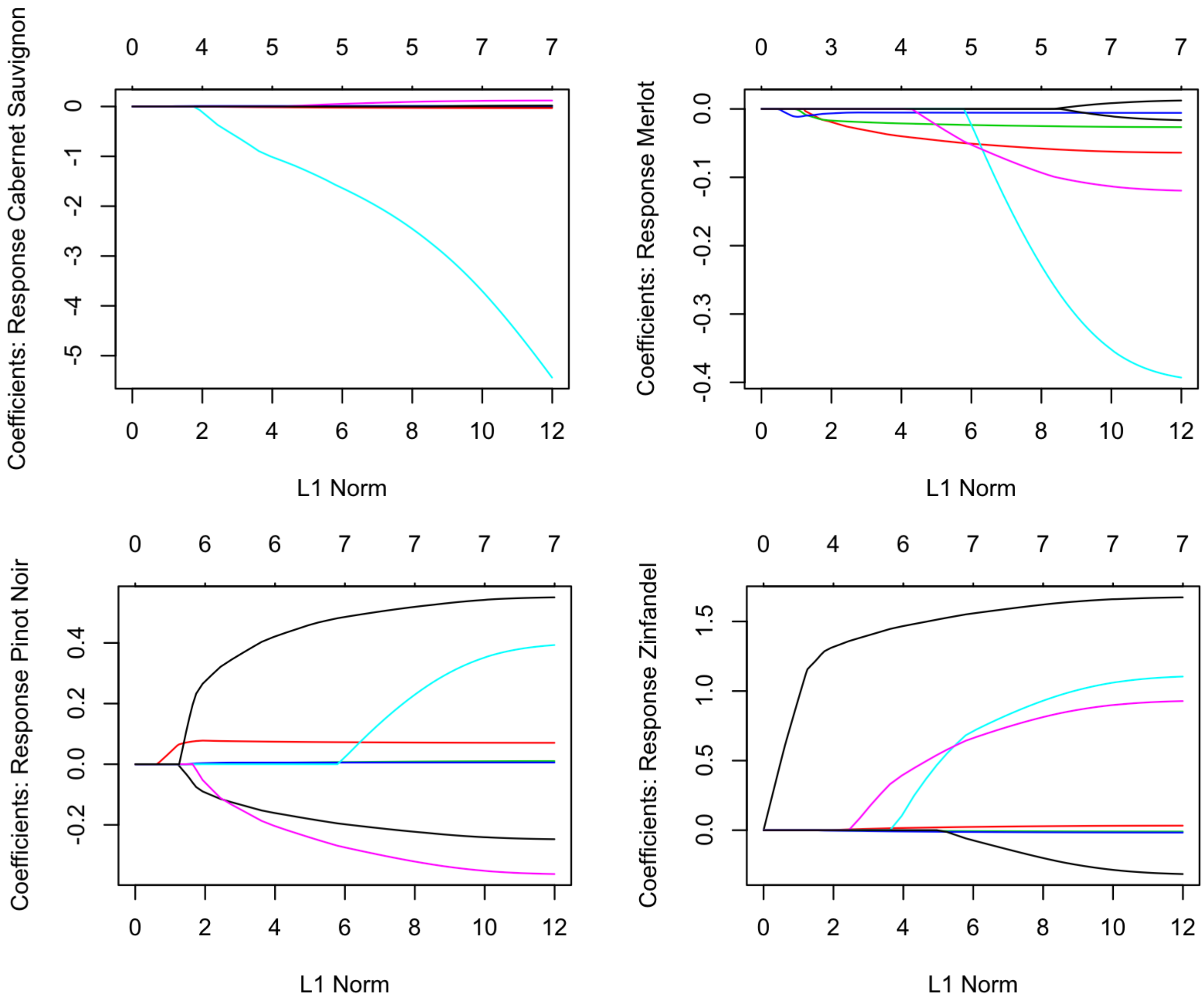
Inspect four linked models, one for each variety

Wharton
Department of Statistics

Fishbone Plots

Lasso paths for the component models

relevant effects
vary over the
models



Coefficients

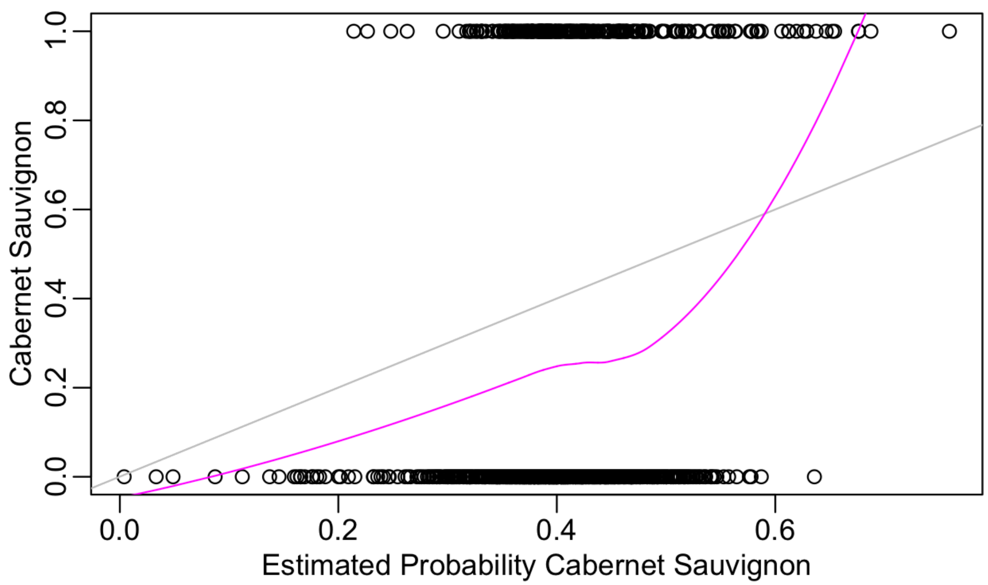
At moderate shrinkage, very different estimates evident for the different varieties

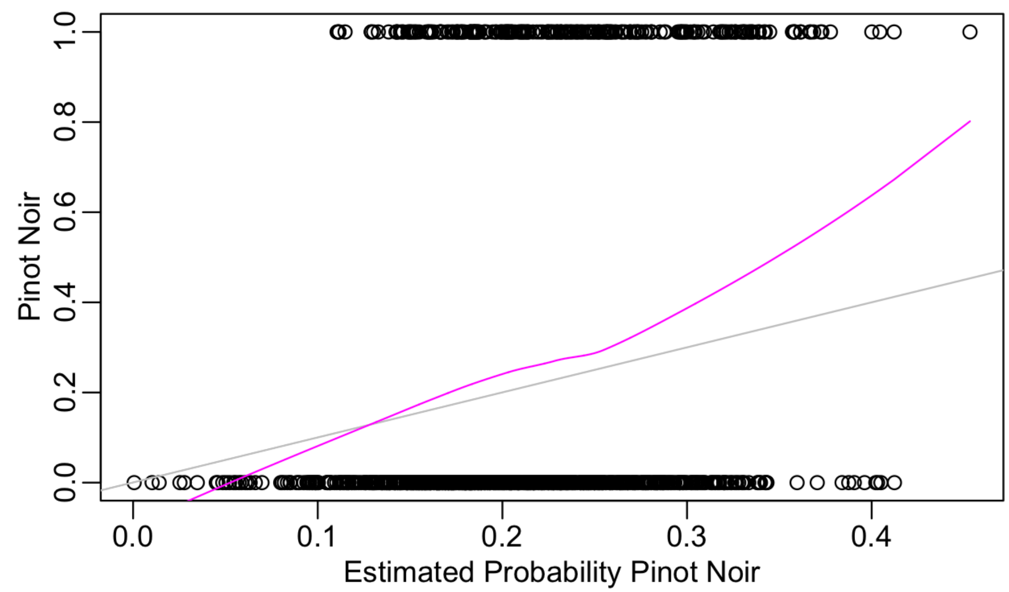
- Need to choose optimal shrinkage
- Relatively dense model with 7 estimates reduced to zero

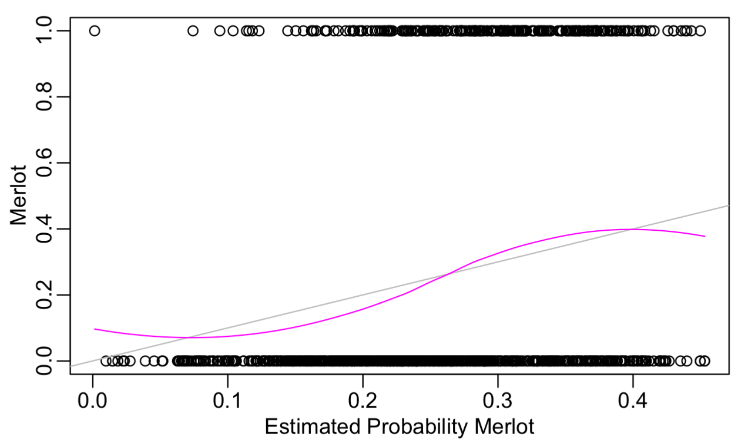
	Cabernet Sauvignon	Merlot	Pinot Noir	Zinfandel
Intercept	56.2284	106.6993	-125.7603	-37.1675
alcohol	0.0000	0.0000	-0.1707	1.4903
vintage	-0.0180	-0.0428	0.0738	0.0180
price	0.0049	-0.0217	0.0051	-0.0049
lengths	0.0103	-0.0054	0.0054	-0.0103
Miss.alcohol	-1.1442	0.0000	0.0000	0.3128
Miss.vintage	0.0064	-0.0064	-0.2229	0.4725
Miss.price	0.0000	0.0000	0.4404	0.0000

Calibration

Models for different varieties are not well calibrated







Merlot model is better calibrated, but also not very high probabilities

Classification Results

Classifier accuracy… not very good

1000 test cases, 250 of each

Easy to get 25% correct without even trying!

Calls most things Cabernet

For example, it correctly identifies only 10 of the Pinots, labeling 230 Pinots as Cabernet.

```
                    multinom.pred
                     Cabernet Sauvignon Merlot Pinot Noir Zinfandel
Cabernet Sauvignon                  219     20          6         5
Merlot                              199     40          6         5
Pinot Noir                          230      6         10         4
Zinfandel                           168     23          0        59
```

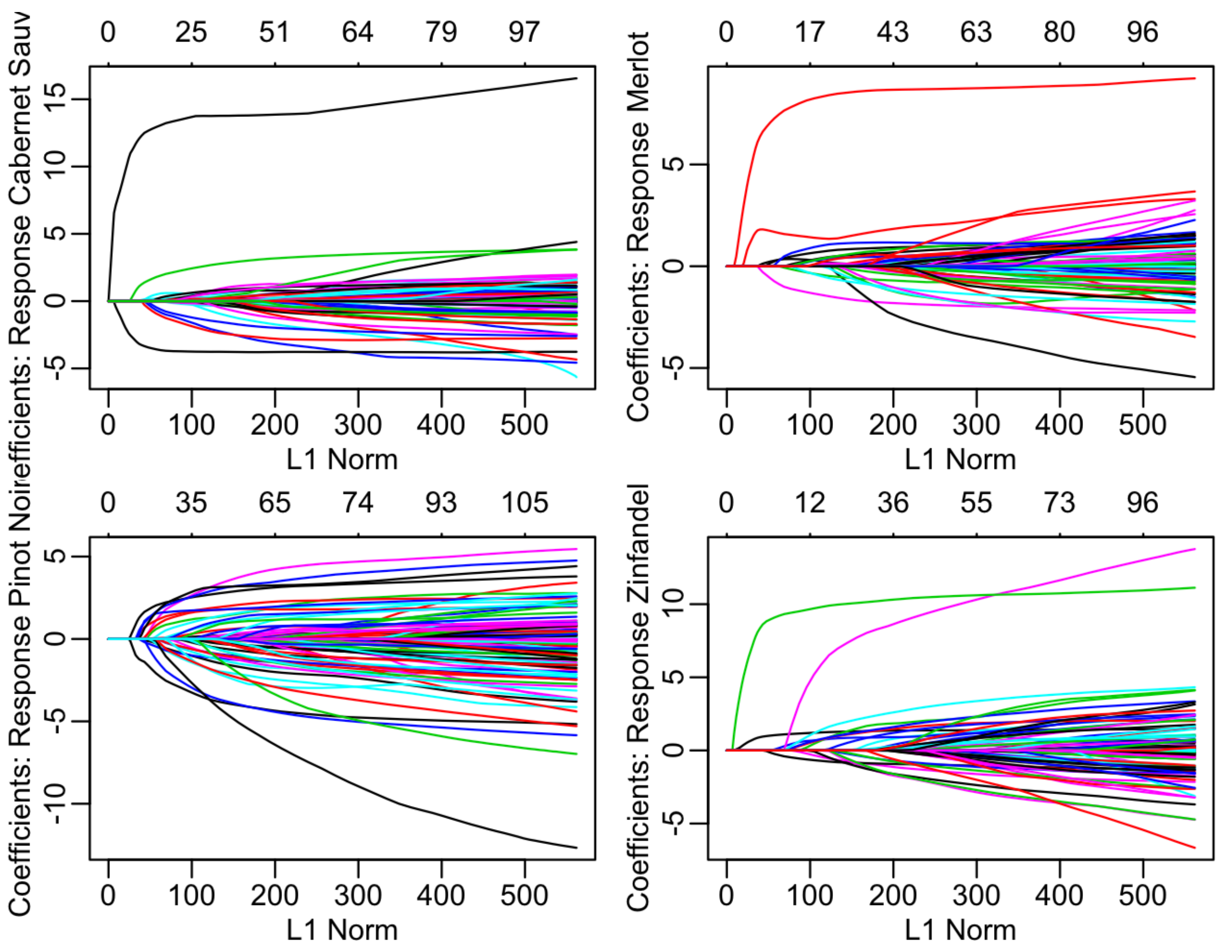
correct = 219+40+10+59 = 329

Add Words

First 100 words

Most common 100 word types

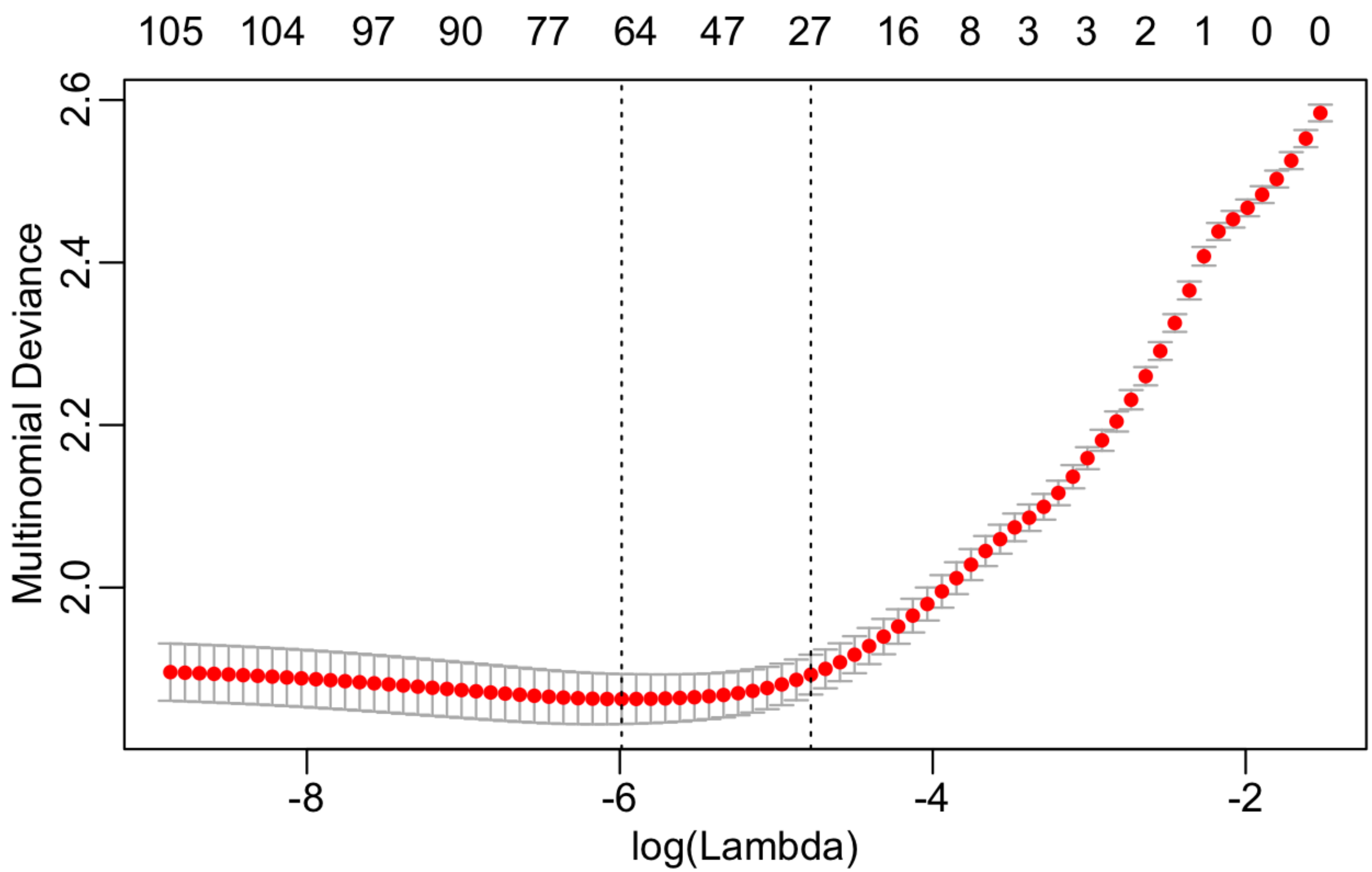
Many more “active” features in models



Wharton
Department of Statistics

Cross-Validate to Tune

Pick tuning parameter from 10-fold CV



Key Words

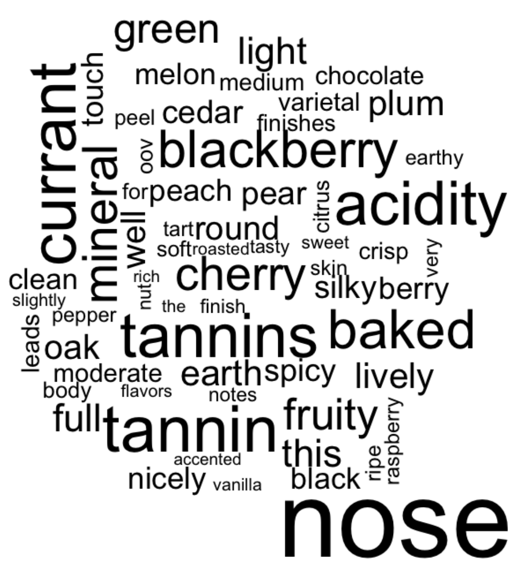
At optimal choice for shrinkage parameter…

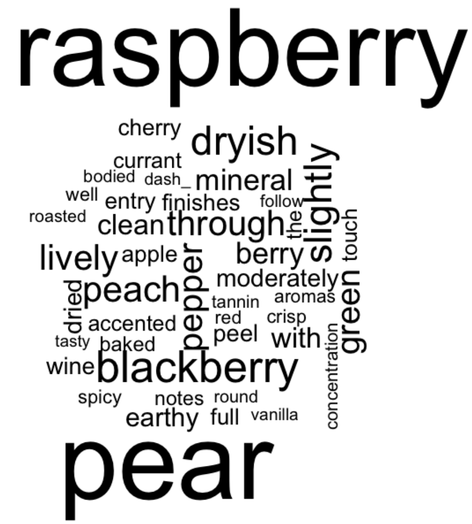
	Cab	Merlot	Pinot	Zin
Cherry	-4	1	3	-1
Currant	14			
Plum		8		
Raspberry				10
Tannin/s			-4	
Pear				6

Cloud View of Coefs

Scaled within each model

beware of warnings





Classification

Much more accurate than baseline model

Accuracy increases from 33% correct to
191 + 133 + 145 + 103 = 572 –> 57% correct

Zinfandel is least accurate, plus fewest in training data

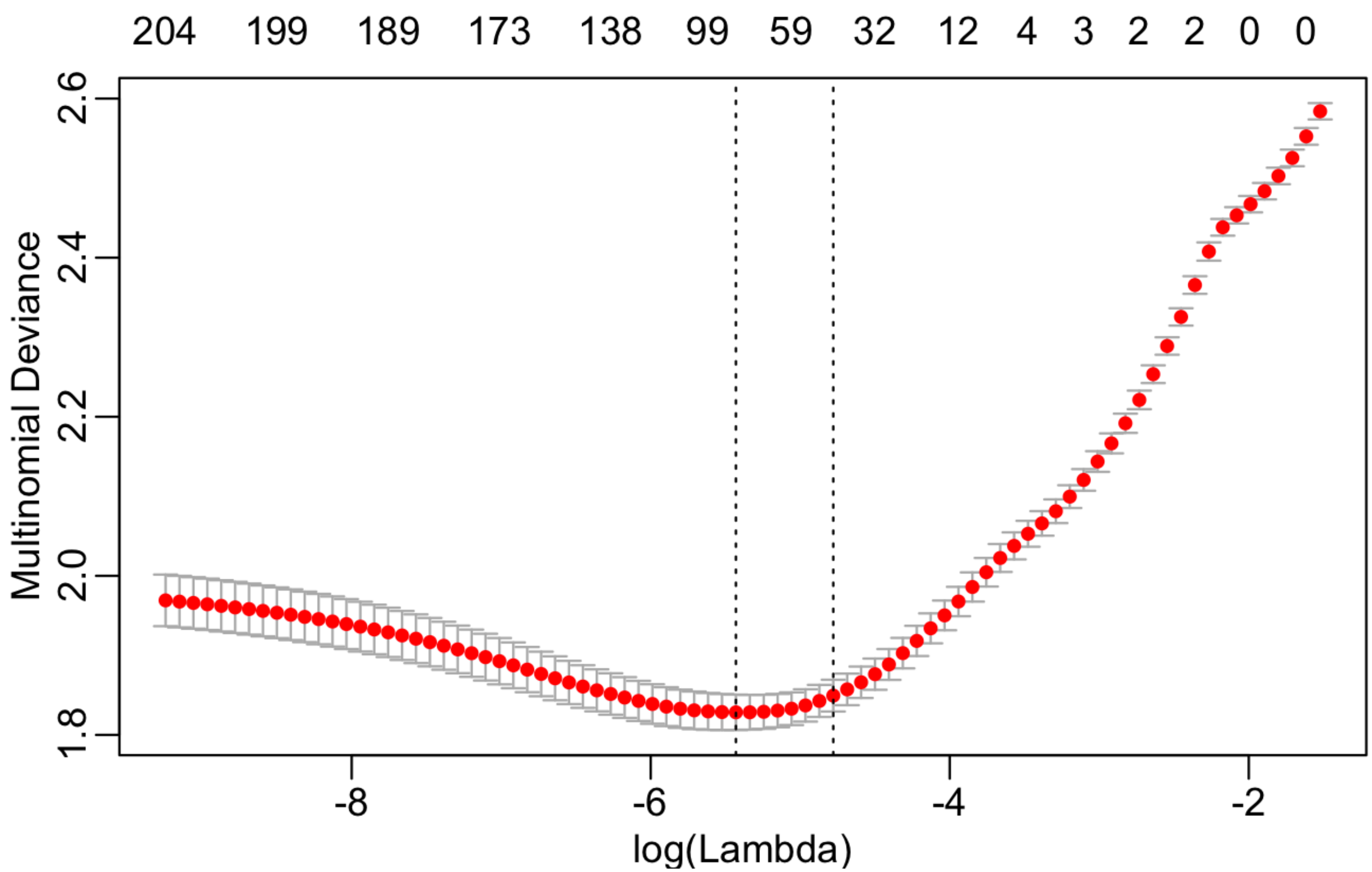
Still tend to classify too many as cabernet… which happens to be most common in the training data!

	Cabernet Sauvignon	Merlot	Pinot Noir	Zinfandel
Cabernet Sauvignon	191	31	22	6
Merlot	61	133	47	9
Pinot Noir	61	34	145	10
Zinfandel	84	30	33	103

Increase to 200 Words

Choice of shrinkage parameter very clear

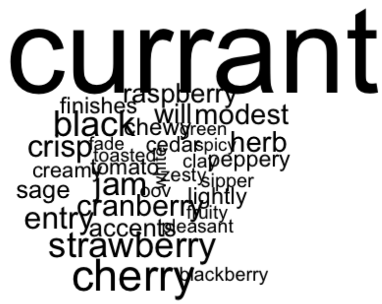
Evident trough indicating best choice for λ

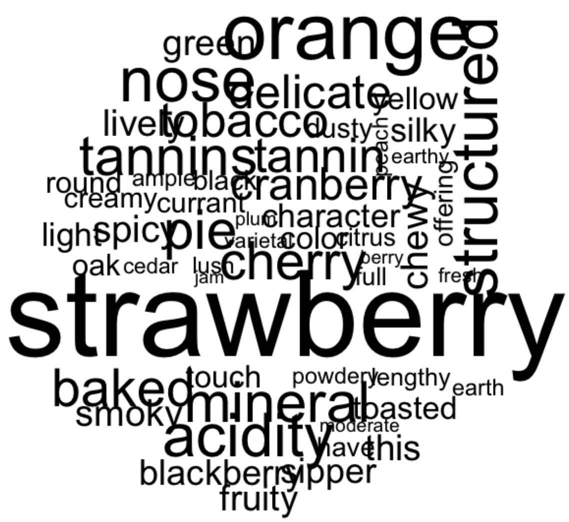


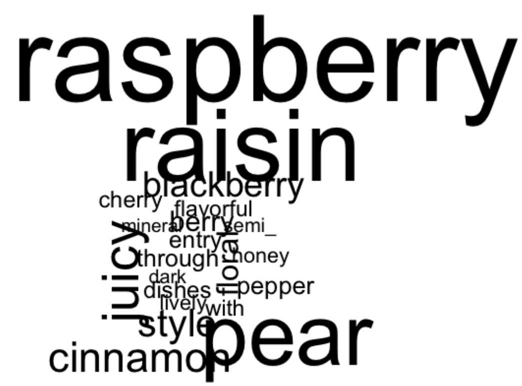
Wharton
Department of Statistics

Coefficient Clouds

Several new terms not available to prior model







Wharton
Department of Statistics

Classification

Not much different from prior model (57% correct)

with 100 words

	Cabernet Sauvignon	Merlot	Pinot Noir	Zinfandel
Cabernet Sauvignon	191	31	22	6
Merlot	61	133	47	9
Pinot Noir	61	34	145	10
Zinfandel	84	30	33	103

with 200 words

multinom.class

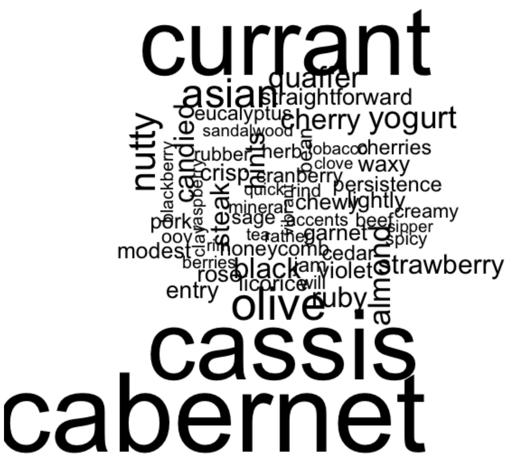
	Cabernet Sauvignon	Merlot	Pinot Noir	Zinfandel
Cabernet Sauvignon	195	33	17	5
Merlot	74	134	35	7
Pinot Noir	64	43	136	7
Zinfandel	86	30	28	106

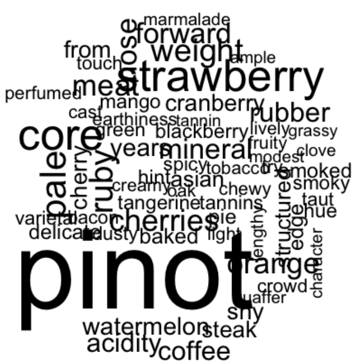
Go Further?

- Lots more words to try
- Tried with 400 words
  - Takes quite a bit longer to run, but works. Again clear trough
  - Some new word types appear… looks like we need to be more careful with preparing our data (next slide)
- Plus, have not explore the importance of combinations of words
  - 2500 words –> 3,125,000 possible (though many would be 0)
  - Other features based on the words present

New Words?

Surprise, surprise!





Classification

No surprising either, this gets better

Percent correct up from 57% to 64%

multinom.class

	Cabernet Sauvignon	Merlot	Pinot Noir	Zinfandel
Cabernet Sauvignon	204	27	12	7
Merlot	60	160	21	9
Pinot Noir	46	33	163	8
Zinfandel	64	39	32	115

What about all of the other words that are available?

Results for Trees

Resemble those obtained from multinomial regression…

See the associated commands in the R script.