Political Science 452: Text as Data

Justin Grimmer

Assistant Professor
Department of Political Science
Stanford University

May 25th, 2011

Where We've Been, Where We're Going

- Class 1: Finding Text Data
- Class 2: Representing Texts Quantitatively
- Class 3: Dictionary Methods for Classification
- Class 4: Comparing Language Across Groups
- Class 5: Texts in Space
- Class 6: Clustering
- Class 7: Topic models
- Class 8: Supervised methods for classification
- Class 9: Ensemble methods for classification
- Class 10: Scaling Speech

Supervised Learning

Week 6 and Week 7:

- Models for discovery
  - Infer categories
  - Infer document assignment to categories
  - Pre-estimation: relatively little work
  - Post-estimation: extensive validation testing

Supervised Learning

Week 6 and Week 7:

- Models for discovery
  - Infer categories
  - Infer document assignment to categories
  - Pre-estimation: relatively little work
  - Post-estimation: extensive validation testing

Week 8 and Week 9:

Supervised Learning

Week 6 and Week 7:

- Models for discovery
  - Infer categories
  - Infer document assignment to categories
  - Pre-estimation: relatively little work
  - Post-estimation: extensive validation testing

Week 8 and Week 9:

- Models for categorizing texts

Supervised Learning

Week 6 and Week 7:

- Models for discovery
  - Infer categories
  - Infer document assignment to categories
  - Pre-estimation: relatively little work
  - Post-estimation: extensive validation testing

Week 8 and Week 9:

- Models for categorizing texts
  - Know (develop) categories before hand

Supervised Learning

Week 6 and Week 7:

- Models for discovery
  - Infer categories
  - Infer document assignment to categories
  - Pre-estimation: relatively little work
  - Post-estimation: extensive validation testing

Week 8 and Week 9:

- Models for categorizing texts
  - Know (develop) categories before hand
  - Hand coding: assign documents to categories
  - Infer: new document assignment to categories (distribution of documents to categories)

Supervised Learning

Week 6 and Week 7:

- Models for discovery
  - Infer categories
  - Infer document assignment to categories
  - Pre-estimation: relatively little work
  - Post-estimation: extensive validation testing

Week 8 and Week 9:

- Models for categorizing texts
  - Know (develop) categories before hand
  - Hand coding: assign documents to categories
  - Infer: new document assignment to categories (distribution of documents to categories)
  - Pre-estimation: extensive work constructing categories, building classifiers
  - Post-estimation: relatively little work

Supervised Learning

This week:

Supervised Learning

This week:

- How to generate valid hand coding categories

Supervised Learning

This week:

- How to generate valid hand coding categories
  - Assessing coder performance
  - Assessing disagreement among coders
  - Evidence coders perform well

Supervised Learning

This week:

- How to generate valid hand coding categories
  - Assessing coder performance
  - Assessing disagreement among coders
  - Evidence coders perform well
- Supervised Learning Methods: Naive Bayes and ReadMe

Supervised Learning

This week:

- How to generate valid hand coding categories
  - Assessing coder performance
  - Assessing disagreement among coders
  - Evidence coders perform well
- Supervised Learning Methods: Naive Bayes and ReadMe
- Assessing Model Performance

Supervised Learning

This week:

- How to generate valid hand coding categories
  - Assessing coder performance
  - Assessing disagreement among coders
  - Evidence coders perform well
- Supervised Learning Methods: Naive Bayes and ReadMe
- Assessing Model Performance

Next week:

Supervised Learning

This week:

- How to generate valid hand coding categories
  - Assessing coder performance
  - Assessing disagreement among coders
  - Evidence coders perform well
- Supervised Learning Methods: Naive Bayes and ReadMe
- Assessing Model Performance

Next week:

- Supervised Learning Method: Support Vector Machines

Supervised Learning

This week:

- How to generate valid hand coding categories
  - Assessing coder performance
  - Assessing disagreement among coders
  - Evidence coders perform well
- Supervised Learning Methods: Naive Bayes and ReadMe
- Assessing Model Performance

Next week:

- Supervised Learning Method: Support Vector Machines
- Ensemble methods: combining the results of many supervised algorithms

Supervised Learning

This week:

- How to generate valid hand coding categories
  - Assessing coder performance
  - Assessing disagreement among coders
  - Evidence coders perform well
- Supervised Learning Methods: Naive Bayes and ReadMe
- Assessing Model Performance

Next week:

- Supervised Learning Method: Support Vector Machines
- Ensemble methods: combining the results of many supervised algorithms
- Cross validation:

Supervised Learning

This week:

- How to generate valid hand coding categories
  - Assessing coder performance
  - Assessing disagreement among coders
  - Evidence coders perform well
- Supervised Learning Methods: Naive Bayes and ReadMe
- Assessing Model Performance

Next week:

- Supervised Learning Method: Support Vector Machines
- Ensemble methods: combining the results of many supervised algorithms
- Cross validation:
  - Replicate classification exercise, with data
  - Avoid over training data: Balance bias and variance in model selection
  - Super learning: optimal ensemble methods

Supervised Learning

This week:

- How to generate valid hand coding categories
  - Assessing coder performance
  - Assessing disagreement among coders
  - Evidence coders perform well
- Supervised Learning Methods: Naive Bayes and ReadMe
- Assessing Model Performance

Next week:

- Supervised Learning Method: Support Vector Machines
- Ensemble methods: combining the results of many supervised algorithms
- Cross validation:
  - Replicate classification exercise, with data
  - Avoid over training data: Balance bias and variance in model selection
  - Super learning: optimal ensemble methods

Methods generalize beyond text

Components to Supervised Learning Method

Components to Supervised Learning Method

1) Set of categories

Components to Supervised Learning Method

1) Set of categories
   - Credit Claiming, Position Taking, Advertising
   - Positive Tone, Negative Tone
   - Pro-war, Ambiguous, Anti-war

Components to Supervised Learning Method

1) Set of categories
   - Credit Claiming, Position Taking, Advertising
   - Positive Tone, Negative Tone
   - Pro-war, Ambiguous, Anti-war
2) Set of hand-coded documents

Components to Supervised Learning Method

1) Set of categories
   - Credit Claiming, Position Taking, Advertising
   - Positive Tone, Negative Tone
   - Pro-war, Ambiguous, Anti-war
2) Set of hand-coded documents
   - Coding done by human coders
   - Training Set: documents we'll use to learn how to code
   - Validation Set: documents we'll use to learn how well we code

Components to Supervised Learning Method

1) Set of categories
   - Credit Claiming, Position Taking, Advertising
   - Positive Tone, Negative Tone
   - Pro-war, Ambiguous, Anti-war
2) Set of hand-coded documents
   - Coding done by human coders
   - Training Set: documents we'll use to learn how to code
   - Validation Set: documents we'll use to learn how well we code
3) Set of unlabeled documents

Components to Supervised Learning Method

1) Set of categories
   - Credit Claiming, Position Taking, Advertising
   - Positive Tone, Negative Tone
   - Pro-war, Ambiguous, Anti-war
2) Set of hand-coded documents
   - Coding done by human coders
   - Training Set: documents we'll use to learn how to code
   - Validation Set: documents we'll use to learn how well we code
3) Set of unlabeled documents
4) Method to extrapolate from hand coding to unlabeled documents

How Do We Generate Coding Rules and Categories?

How Do We Generate Coding Rules and Categories?

Challenge: coding rules/training coders to maximize coder performance

How Do We Generate Coding Rules and Categories?

Challenge: coding rules/training coders to maximize coder performance
Challenge: developing a clear set of categories

How Do We Generate Coding Rules and Categories?

Challenge: coding rules/training coders to maximize coder performance

Challenge: developing a clear set of categories

1) Limits of Humans:

How Do We Generate Coding Rules and Categories?

Challenge: coding rules/training coders to maximize coder performance

Challenge: developing a clear set of categories

1) Limits of Humans:
   - Small working memories
   - Easily distracted
   - Insufficient motivation

How Do We Generate Coding Rules and Categories?

Challenge: coding rules/training coders to maximize coder performance

Challenge: developing a clear set of categories

1) Limits of Humans:
   - Small working memories
   - Easily distracted
   - Insufficient motivation
2) Limits of Language:

How Do We Generate Coding Rules and Categories?

Challenge: coding rules/training coders to maximize coder performance

Challenge: developing a clear set of categories

1) Limits of Humans:
   - Small working memories
   - Easily distracted
   - Insufficient motivation
2) Limits of Language:
   - Fundamental ambiguity in language [careful analysis of texts]
   - Contextual nature of language

How Do We Generate Coding Rules and Categories?

Challenge: coding rules/training coders to maximize coder performance

Challenge: developing a clear set of categories

1) Limits of Humans:
   - Small working memories
   - Easily distracted
   - Insufficient motivation
2) Limits of Language:
   - Fundamental ambiguity in language [careful analysis of texts]
   - Contextual nature of language

For supervised methods to work: maximize coder agreement

How Do We Generate Coding Rules and Categories?

Challenge: coding rules/training coders to maximize coder performance

Challenge: developing a clear set of categories

1) Limits of Humans:
   - Small working memories
   - Easily distracted
   - Insufficient motivation
2) Limits of Language:
   - Fundamental ambiguity in language [careful analysis of texts]
   - Contextual nature of language

For supervised methods to work: maximize coder agreement

1) Write careful (and brief) coding rules

How Do We Generate Coding Rules and Categories?

Challenge: coding rules/training coders to maximize coder performance

Challenge: developing a clear set of categories

1) Limits of Humans:
   - Small working memories
   - Easily distracted
   - Insufficient motivation
2) Limits of Language:
   - Fundamental ambiguity in language [careful analysis of texts]
   - Contextual nature of language

For supervised methods to work: maximize coder agreement

1) Write careful (and brief) coding rules
   - Flow charts help simplify problems

How Do We Generate Coding Rules and Categories?

Challenge: coding rules/training coders to maximize coder performance

Challenge: developing a clear set of categories

1) Limits of Humans:
   - Small working memories
   - Easily distracted
   - Insufficient motivation
2) Limits of Language:
   - Fundamental ambiguity in language [careful analysis of texts]
   - Contextual nature of language

For supervised methods to work: maximize coder agreement

1) Write careful (and brief) coding rules
   - Flow charts help simplify problems
2) Train coders to remove ambiguity, misinterpretation

How Do We Generate Coding Rules?

Iterative process for generating coding rules:

How Do We Generate Coding Rules?

Iterative process for generating coding rules:

1) Write a set of coding rules

How Do We Generate Coding Rules?

Iterative process for generating coding rules:

1) Write a set of coding rules
2) Have coders code documents (about 200)

How Do We Generate Coding Rules?

Iterative process for generating coding rules:

1) Write a set of coding rules
2) Have coders code documents (about 200)
3) Assess coder agreement

How Do We Generate Coding Rules?

Iterative process for generating coding rules:

1) Write a set of coding rules
2) Have coders code documents (about 200)
3) Assess coder agreement
4) Identify sources of disagreement, repeat

How Do We Identify Coding Disagreement?

Many measures of inter-coder agreement

Essentially attempt to summarize a confusion matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Sum, Coder 1
Cat 1	**30**	0	1	0	31
Cat 2	1	**1**	0	0	2
Cat 3	0	0	**1**	0	1
Cat 4	3	1	0	**7**	11
Sum, Coder 2	34	2	2	7	Total: **45**

- **Diagonal**: coders agree on document
- Off-diagonal : coders disagree (confused) on document

Generalize across (k) coders:

- $\frac{k(k-1)}{2}$ pairwise comparisons
- k comparisons: Coder A against All other coders

How Do We Identify Coding Disagreements?

During coding development phase/coder assessment phase, full confusion matrices help to identify

- Ambiguity
- Coder slacking

Example: 3 Coders, 8 categories.

How Do We Identify Coding Disagreements?

During coding development phase/coder assessment phase, full confusion matrices help to identify

- Ambiguity
- Coder slacking

Example: 3 Coders, 8 categories.

	Coder A								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Tot**
Coder B									
1	**15**	2	1	0	0	1	0	0	
3	1	0	**0**	1	0	0	0	0	
4	0	0	0	**5**	0	3	1	0	
5	0	0	0	1	**13**	7	0	2	
6	11	1	3	3	1	**32**	0	1	
7	1	0	0	0	0	13	**26**	36	
8	2	0	0	0	1	7	0	**8**	
Total	30	3	4	10	15	63	27	47	

How Do We Identify Coding Disagreements?

During coding development phase/coder assessment phase, full confusion matrices help to identify

- Ambiguity
- Coder slacking

Example: 3 Coders, 8 categories.

				Coder A					
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	Tota
Coder C									
1	**23**	1	1	1	0	9	0	0	
2	0	0	0	0	0	1	0	0	
3	1	1	**3**	2	0	3	0	0	
4	0	0	0	**4**	0	8	1	0	
5	0	0	0	2	**13**	2	0	2	
6	4	1	0	1	1	**32**	1	2	
7	1	0	0	0	0	2	**25**	36	
8	1	0	0	0	1	6	0	**7**	
Total	30	3	4	10	15	63	27	47	

How Do We Identify Coding Disagreements?

During coding development phase/coder assessment phase, full confusion matrices help to identify

- Ambiguity
- Coder slacking

Example: 3 Coders, 8 categories.

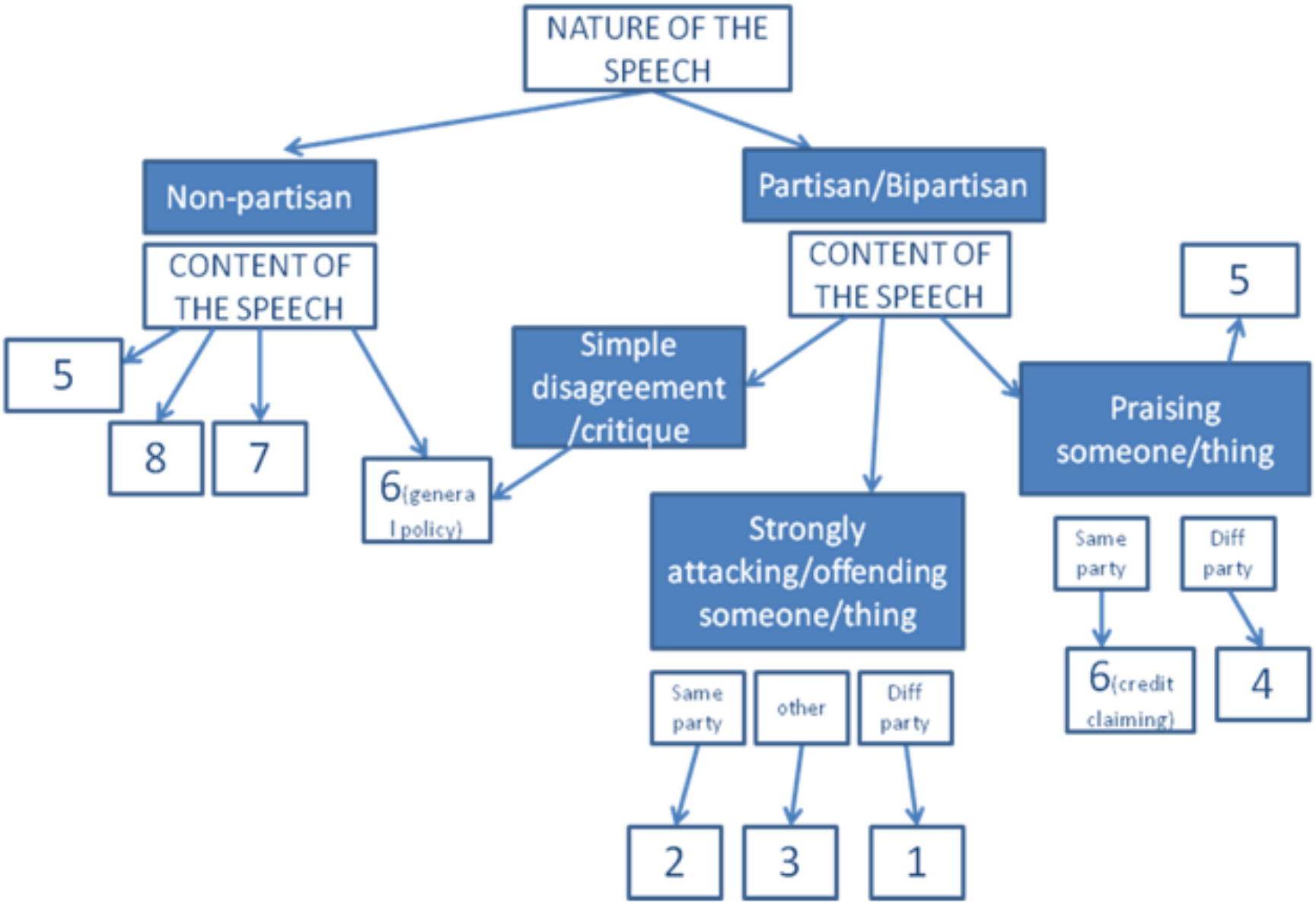
			Coder C						
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	Tota
Coder B									
1	**18**	0	1	0	0	0	0	0	
3	1	0	**1**	0	0	0	0	0	
4	0	0	1	**7**	0	1	0	0	
5	0	0	0	2	**18**	3	0	0	
6	13	1	7	4	1	**26**	0	0	
7	3	0	0	0	0	8	**63**	2	
8	0	0	0	0	0	4	1	**15**	
Total	35	1	10	13	19	42	64	17	

Example Coding Document

8 part coding scheme

- Across Party Taunting: explicit public and negative attacks on the other party or its members
- Within Party Taunting: explicit public and negative attacks on the same party or its members [for 1960's politics]
- Other taunting: explicit public and negative attacks not directed at a party
- Bipartisan support: praise for the other party
- Honorary Statements: qualitatively different kind of speech
- Policy speech: a speech without taunting or credit claiming
- Procedural
- No Content: (occasionally occurs in CR)

Example Coding Document



How Do We Summarize Confusion Matrix?

Lots of statistics to summarize confusion matrix:

- Most common: intercoder agreement

$$\text{Inter Coder}(A, B) = \frac{\text{No. (Coder A \& Coder B agree)}}{\text{No. Documents}}$$

Liberal measure of agreement:

Liberal measure of agreement:

- Some agreement by chance

Liberal measure of agreement:

- Some agreement by chance
- Consider coding scheme with two categories $\{$ Class 1, Class 2$\}$.

Liberal measure of agreement:

- Some agreement by chance
- Consider coding scheme with two categories $\{$ Class 1, Class 2$\}$.
- Coder A and Coder B flip a (biased coin).
  (Pr(Class 1) $= 0.75$, Pr(Class 2) $= 0.25$)

Liberal measure of agreement:

- Some agreement by chance
- Consider coding scheme with two categories $\{$ Class 1, Class 2$\}$.
- Coder A and Coder B flip a (biased coin). (Pr(Class 1) $= 0.75$, Pr(Class 2) $= 0.25$)
- Inter Coder reliability: 0.625

Liberal measure of agreement:

- Some agreement by chance
- Consider coding scheme with two categories $\{$ Class 1, Class 2$\}$.
- Coder A and Coder B flip a (biased coin). (Pr(Class 1) $= 0.75$, Pr(Class 2) $= 0.25$)
- Inter Coder reliability: 0.625

What to do?

Liberal measure of agreement:

- Some agreement by chance
- Consider coding scheme with two categories $\{$ Class 1, Class 2$\}$.
- Coder A and Coder B flip a (biased coin). (Pr(Class 1) $= 0.75$, Pr(Class 2) $= 0.25$)
- Inter Coder reliability: 0.625

What to do?

Suggestion: Subtract off amount expected by chance:

Liberal measure of agreement:

- Some agreement by chance
- Consider coding scheme with two categories $\{$ Class 1, Class 2$\}$.
- Coder A and Coder B flip a (biased coin). (Pr(Class 1) $= 0.75$, Pr(Class 2) $= 0.25$)
- Inter Coder reliability: 0.625

What to do?

Suggestion: Subtract off amount expected by chance:

$$\text{Inter Coder}(A, B)_{\text{norm}} = \frac{\text{No. (Coder A \& Coder B agree)} - \text{No. Expected by Chance}}{\text{No. Documents} - \text{No. Expected by Chance}}$$

Liberal measure of agreement:

- Some agreement by chance
- Consider coding scheme with two categories $\{$ Class 1, Class 2$\}$.
- Coder A and Coder B flip a (biased coin). (Pr(Class 1) $= 0.75$, Pr(Class 2) $= 0.25$)
- Inter Coder reliability: 0.625

What to do?

Suggestion: Subtract off amount expected by chance:

$$\text{Inter Coder}(A, B)_{\text{norm}} = \frac{\text{No. (Coder A \& Coder B agree)} - \text{No. Expected by Chance}}{\text{No. Documents} - \text{No. Expected by Chance}}$$

Question: what is amount expected by chance?

Liberal measure of agreement:

- Some agreement by chance
- Consider coding scheme with two categories $\{$ Class 1, Class 2$\}$.
- Coder A and Coder B flip a (biased coin). (Pr(Class 1) $= 0.75$, Pr(Class 2) $= 0.25$)
- Inter Coder reliability: 0.625

What to do?

Suggestion: Subtract off amount expected by chance:

$$\text{Inter Coder}(A, B)_{\text{norm}} = \frac{\text{No. (Coder A \& Coder B agree)} - \text{No. Expected by Chance}}{\text{No. Documents} - \text{No. Expected by Chance}}$$

Question: what is amount expected by chance?

- $\frac{1}{\#\text{Categories}}$?
- Avg Proportion in categories across coders? (Krippendorf's Alpha)

Liberal measure of agreement:

- Some agreement by chance
- Consider coding scheme with two categories $\{$ Class 1, Class 2$\}$.
- Coder A and Coder B flip a (biased coin). (Pr(Class 1) $= 0.75$, Pr(Class 2) $= 0.25$)
- Inter Coder reliability: 0.625

What to do?

Suggestion: Subtract off amount expected by chance:

$$\text{Inter Coder}(A, B)_{\text{norm}} = \frac{\text{No. (Coder A \& Coder B agree)} - \text{No. Expected by Chance}}{\text{No. Documents} - \text{No. Expected by Chance}}$$

Question: what is amount expected by chance?

- $\frac{1}{\#\text{Categories}}$?
- Avg Proportion in categories across coders? (Krippendorf's Alpha)

Best Practice: present confusion matrices.

Krippendorf's Alpha, With Deference to Communication Students a Critique

Define coder reliability as:

Krippendorf's Alpha, With Deference to Communication Students a Critique

Define coder reliability as:

$$\alpha = 1 - \frac{\text{No. Pairwise Disagreements Observed}}{\text{No Pairwise Disagreements Expected By Chance}}$$

Krippendorf's Alpha, With Deference to Communication Students a Critique

Define coder reliability as:

$$\alpha = 1 - \frac{\text{No. Pairwise Disagreements Observed}}{\text{No Pairwise Disagreements Expected By Chance}}$$

No. Pairwise Disagreements Observed = observe from data

Krippendorf's Alpha, With Deference to Communication Students a Critique

Define coder reliability as:

$$\alpha = 1 - \frac{\text{No. Pairwise Disagreements Observed}}{\text{No Pairwise Disagreements Expected By Chance}}$$

No. Pairwise Disagreements Observed = observe from data

No Expected pairwise disagreements: coding by chance, with rate labels used available from data

Krippendorf's Alpha, With Deference to Communication Students a Critique

Define coder reliability as:

$$\alpha = 1 - \frac{\text{No. Pairwise Disagreements Observed}}{\text{No Pairwise Disagreements Expected By Chance}}$$

No. Pairwise Disagreements Observed = observe from data

No Expected pairwise disagreements: coding by chance, with rate labels used available from data

Problem with family of statistics:

Krippendorf's Alpha, With Deference to Communication Students a Critique

Define coder reliability as:

$$\alpha = 1 - \frac{\text{No. Pairwise Disagreements Observed}}{\text{No Pairwise Disagreements Expected By Chance}}$$

No. Pairwise Disagreements Observed = observe from data

No Expected pairwise disagreements: coding by chance, with rate labels used available from data

Problem with family of statistics:

- Pretend I know something I'm trying to estimate
- How is that we know coders estimate levels well?
- Have to present correlation statistic: vary assumptions about "expectations" (from uniform, to data driven)

Krippendorf's Alpha, With Deference to Communication Students a Critique

Define coder reliability as:

$$\alpha = 1 - \frac{\text{No. Pairwise Disagreements Observed}}{\text{No Pairwise Disagreements Expected By Chance}}$$

No. Pairwise Disagreements Observed = observe from data

No Expected pairwise disagreements: coding by chance, with rate labels used available from data

Problem with family of statistics:

- Pretend I know something I'm trying to estimate
- How is that we know coders estimate levels well?
- Have to present correlation statistic: vary assumptions about "expectations" (from uniform, to data driven)

Calculate in `R` with `concord` package and function `kripp.alpha`

How Many To Code By Hand/How Many to Code By Machine

Next week: we'll discuss how to answer this question systematically for your data set.

Rules of thumb:

- Hopkins and King (2010): 500 documents likely sufficient
- Hopkins and King (2010): 100 documents may be enough
- BUT: depends on quantity of interest
- May REQUIRE many more documents

Percent data coded, Error (From Dan Jurafsky)

Training size

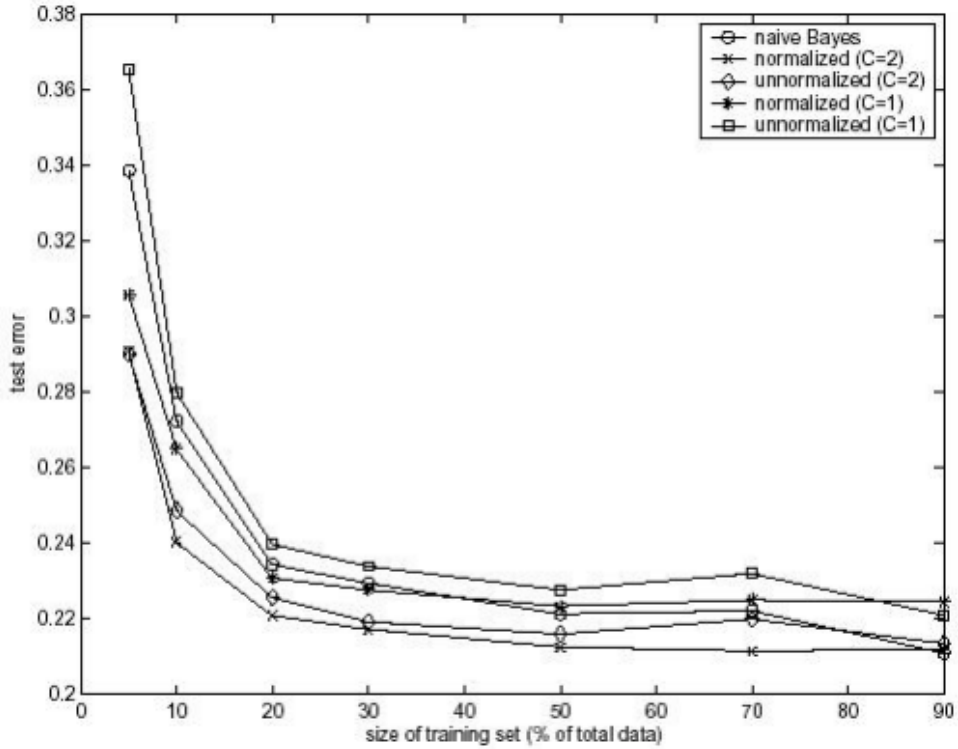


Figure 2: Test error vs training size on the newsgroups alt.atheism and talk.religion.misc

Three categories of documents

Hand labeled

- Training set (what we'll use to estimate model)
- Validation set (what we'll use to assess model)

Unlabeled

- Test set (what we'll use the model to categorize)

Label more documents than necessary to train model

Methods to Perform Supervised Classification

- Naive Bayes

Methods to Perform Supervised Classification

- Naive Bayes
- Support Vector Machines (Introduce Week 9, with Cross validation and Ensembles)

Methods to Perform Supervised Classification

- Naive Bayes
- Support Vector Machines (Introduce Week 9, with Cross validation and Ensembles)
- ReadMe (optimized for a different objective)

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with M features

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with M features $\mathbf{y}_i = (y_{1i}, y_{2i}, \ldots, y_{Mi})$

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with M features
$\mathbf{y}_i = (y_{1i}, y_{2i}, \ldots, y_{Mi})$
Set of J categories. Category j $(j = 1, \ldots, J)$
$\{C_1, C_2, \ldots, C_J\}$

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with M features
$\mathbf{y}_i = (y_{1i}, y_{2i}, \ldots, y_{Mi})$
Set of J categories. Category j $(j = 1, \ldots, J)$
$\{C_1, C_2, \ldots, C_J\}$
Goal: classify every document into one category.

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with M features
$\mathbf{y}_i = (y_{1i}, y_{2i}, \ldots, y_{Mi})$
Set of J categories. Category j $(j = 1, \ldots, J)$
$\{C_1, C_2, \ldots, C_J\}$
Goal: classify every document into one category.
Learn a function that maps from space of (possible) documents to categories

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with M features
$\mathbf{y}_i = (y_{1i}, y_{2i}, \ldots, y_{Mi})$
Set of J categories. Category j $(j = 1, \ldots, J)$
$\{C_1, C_2, \ldots, C_J\}$
Goal: classify every document into one category.
Learn a function that maps from space of (possible) documents to categories
To do this: use hand coded observations to estimate (train) regression model

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with M features
$\mathbf{y}_i = (y_{1i}, y_{2i}, \ldots, y_{Mi})$
Set of J categories. Category j $(j = 1, \ldots, J)$
$\{C_1, C_2, \ldots, C_J\}$
Goal: classify every document into one category.
Learn a function that maps from space of (possible) documents to categories
To do this: use hand coded observations to estimate (train) regression model
Apply model to test data, classify those observations

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Goal: For each document $\mathbf{y}_i$, we want to infer most likely category

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Goal: For each document $\mathbf{y}_i$, we want to infer most likely category

$$C_{\text{Max}} = \arg\max_j p(C_j|\mathbf{y}_i)$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Goal: For each document $\mathbf{y}_i$, we want to infer most likely category

$$C_{\text{Max}} = \arg\max_j p(C_j|\mathbf{y}_i)$$

We're going to use Bayes' rule to estimate $p(C_j|\mathbf{y}_i)$.

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Goal: For each document $\mathbf{y}_i$, we want to infer most likely category

$$C_{\text{Max}} = \arg\max_j p(C_j|\mathbf{y}_i)$$

We're going to use Bayes' rule to estimate $p(C_j|\mathbf{y}_i)$.

$$\begin{aligned} p(C_j|\mathbf{y}_i) &= \frac{p(C_j, \mathbf{y}_i)}{p(\mathbf{y}_i)} \\ &= \frac{p(C_j)p(\mathbf{y}_i|C_j)}{p(\mathbf{y}_i)} \end{aligned}$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

$$C_{\text{Max}} = \arg\max_j p(C_j|\mathbf{y}_i)$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_j \; p(C_j|\mathbf{y}_i) \\
C_{\text{Max}} &= \arg\max_j \; \frac{p(C_j)p(\mathbf{y}_i|C_j)}{p(\mathbf{y}_i)}
\end{aligned}
$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_j \; p(C_j|\mathbf{y}_i) \\
C_{\text{Max}} &= \arg\max_j \; \frac{p(C_j)p(\mathbf{y}_i|C_j)}{p(\mathbf{y}_i)} \\
C_{\text{Max}} &= \arg\max_j \; p(C_j)p(\mathbf{y}_i|C_j)
\end{aligned}
$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_j \; p(C_j|\mathbf{y}_i) \\
C_{\text{Max}} &= \arg\max_j \; \frac{p(C_j)p(\mathbf{y}_i|C_j)}{p(\mathbf{y}_i)} \\
C_{\text{Max}} &= \arg\max_j \; p(C_j)p(\mathbf{y}_i|C_j)
\end{aligned}
$$

Two probabilities to estimate:

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_j \; p(C_j|\mathbf{y}_i) \\
C_{\text{Max}} &= \arg\max_j \; \frac{p(C_j)p(\mathbf{y}_i|C_j)}{p(\mathbf{y}_i)} \\
C_{\text{Max}} &= \arg\max_j \; p(C_j)p(\mathbf{y}_i|C_j)
\end{aligned}
$$

Two probabilities to estimate:

$$p(C_j) = \frac{\text{No. Documents in } j}{\text{No. Documents}} \text{ (training set)}$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_j \; p(C_j|\mathbf{y}_i) \\
C_{\text{Max}} &= \arg\max_j \; \frac{p(C_j)p(\mathbf{y}_i|C_j)}{p(\mathbf{y}_i)} \\
C_{\text{Max}} &= \arg\max_j \; p(C_j)p(\mathbf{y}_i|C_j)
\end{aligned}
$$

Two probabilities to estimate:

$p(C_j) = \frac{\text{No. Documents in } j}{\text{No. Documents}}$ (training set)

$p(\mathbf{y}_i|C_j)$ complicated without assumptions

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_j \; p(C_j|\mathbf{y}_i) \\
C_{\text{Max}} &= \arg\max_j \; \frac{p(C_j)p(\mathbf{y}_i|C_j)}{p(\mathbf{y}_i)} \\
C_{\text{Max}} &= \arg\max_j \; p(C_j)p(\mathbf{y}_i|C_j)
\end{aligned}
$$

Two probabilities to estimate:

$p(C_j) = \frac{\text{No. Documents in } j}{\text{No. Documents}}$ (training set)

$p(\mathbf{y}_i|C_j)$ complicated without assumptions

- Imagine each y_{im} just binary indicator. Then 2^M possible $\mathbf{y}_i$ documents

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_j \; p(C_j|\mathbf{y}_i) \\
C_{\text{Max}} &= \arg\max_j \; \frac{p(C_j)p(\mathbf{y}_i|C_j)}{p(\mathbf{y}_i)} \\
C_{\text{Max}} &= \arg\max_j \; p(C_j)p(\mathbf{y}_i|C_j)
\end{aligned}
$$

Two probabilities to estimate:

$p(C_j) = \frac{\text{No. Documents in } j}{\text{No. Documents}}$ (training set)

$p(\mathbf{y}_i|C_j)$ complicated without assumptions

- Imagine each y_{im} just binary indicator. Then 2^M possible $\mathbf{y}_i$ documents
- Simplify: assume each feature is independent

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_j \; p(C_j|\mathbf{y}_i) \\
C_{\text{Max}} &= \arg\max_j \; \frac{p(C_j)p(\mathbf{y}_i|C_j)}{p(\mathbf{y}_i)} \\
C_{\text{Max}} &= \arg\max_j \; p(C_j)p(\mathbf{y}_i|C_j)
\end{aligned}
$$

Two probabilities to estimate:

$p(C_j) = \frac{\text{No. Documents in } j}{\text{No. Documents}}$ (training set)

$p(\mathbf{y}_i|C_j)$ complicated without assumptions

- Imagine each y_{im} just binary indicator. Then 2^M possible $\mathbf{y}_i$ documents
- Simplify: assume each feature is independent

$$
p(\mathbf{y}_i|C_j) = \prod_{m=1}^{M} p(y_{im}|C_j)
$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Two components to estimation:

- $p(C_j) = \frac{\text{No. Documents in } j}{\text{No. Documents}}$ (training set)
- $p(\mathbf{y}_i|C_j) = \prod_{m=1}^{M} p(y_{im}|C_j)$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Two components to estimation:

- $p(C_j) = \frac{\text{No. Documents in } j}{\text{No. Documents}}$ (training set)
- $p(\mathbf{y}_i|C_j) = \prod_{m=1}^{M} p(y_{im}|C_j)$

Maximum likelihood estimation (training set):

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Two components to estimation:

- $p(C_j) = \frac{\text{No. Documents in } j}{\text{No. Documents}}$ (training set)
- $p(\mathbf{y}_i|C_j) = \prod_{m=1}^{M} p(y_{im}|C_j)$

Maximum likelihood estimation (training set):

$$p(y_{im} = x|C_j) \quad = \quad \frac{\text{No(Docs}_{im} = \text{x and C} = \text{C}_j \text{)}}{\text{No(C= C}_j\text{)}}$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Two components to estimation:

- $p(C_j) = \frac{\text{No. Documents in } j}{\text{No. Documents}}$ (training set)
- $p(\mathbf{y}_i|C_j) = \prod_{m=1}^{M} p(y_{im}|C_j)$

Maximum likelihood estimation (training set):

$$p(y_{im} = x|C_j) = \frac{\text{No(Docs}_{im} = \text{x and C} = \text{C}_j \text{)}}{\text{No(C= C}_j\text{)}}$$

Problem: What if No($\text{Docs}_{im} = \text{x and C} = \text{C}_j$) $= 0$?

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Two components to estimation:

- $p(C_j) = \frac{\text{No. Documents in } j}{\text{No. Documents}}$ (training set)
- $p(\mathbf{y}_i|C_j) = \prod_{m=1}^{M} p(y_{im}|C_j)$

Maximum likelihood estimation (training set):

$$p(y_{im} = x|C_j) \quad = \quad \frac{\text{No(Docs}_{im} = \text{x and C} = \text{C}_j \text{)}}{\text{No(C= C}_j\text{)}}$$

Problem: What if No($\text{Docs}_{im} = \text{x and C} = \text{C}_j$) $= 0$?
$\prod_{m=1}^{M} p(y_{im}|C_j) = 0$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(y_{im} = x|C_j) \quad = \quad \frac{\text{No}(\ \text{Docs}_{im} = \text{x and C} = \text{C}_j\ ) + 1}{\text{No}(\text{C}= \text{C}_j) + k}$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(y_{im} = x|C_j) \quad = \quad \frac{\text{No}(\ \text{Docs}_{im} = \text{x and C} = \text{C}_j\ ) + 1}{\text{No}(\text{C}= \text{C}_j) + k}$$

Algorithm steps:

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(y_{im} = x|C_j) \quad = \quad \frac{\text{No(Docs}_{im} = \text{x and C} = \text{C}_j \text{)} + 1}{\text{No(C= C}_j) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\mathbf{y}_i|C_j)$ on training data

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(y_{im} = x|C_j) \quad = \quad \frac{\text{No}(\ \text{Docs}_{im} = \text{x and C} = \text{C}_j\ ) + 1}{\text{No}(\text{C}= \text{C}_j) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\mathbf{y}_i|C_j)$ on training data
2) Use this to identify most likely C_j for each document i in test set

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(y_{im} = x|C_j) \quad = \quad \frac{\text{No(Docs}_{im} = \text{x and C} = \text{C}_j \text{)} + 1}{\text{No(C= C}_j) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\mathbf{y}_i|C_j)$ on training data
2) Use this to identify most likely C_j for each document i in test set

$$C_i \quad = \quad \arg\max_j \hat{p}(C_j)\hat{p}(\mathbf{y}_i|C_j)$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(y_{im} = x|C_j) \quad = \quad \frac{\text{No(Docs}_{im} = \text{x and C} = \text{C}_j \text{)} + 1}{\text{No(C= C}_j) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\mathbf{y}_i|C_j)$ on training data
2) Use this to identify most likely C_j for each document i in test set

$$C_i \quad = \quad \text{arg max }_j \hat{p}(C_j)\hat{p}(\mathbf{y}_i|C_j)$$

Simple intuition about Naive Bayes:

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(y_{im} = x|C_j) \quad = \quad \frac{\text{No(Docs}_{im} = \text{x and C} = \text{C}_j \text{)} + 1}{\text{No(C= C}_j) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\mathbf{y}_i|C_j)$ on training data
2) Use this to identify most likely C_j for each document i in test set

$$C_i \quad = \quad \arg\max_j \hat{p}(C_j)\hat{p}(\mathbf{y}_i|C_j)$$

Simple intuition about Naive Bayes:

- Learn what documents in class j look like

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(y_{im} = x|C_j) \quad = \quad \frac{\text{No(Docs}_{im} = \text{x and C} = \text{C}_j \text{)} + 1}{\text{No(C= C}_j) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\mathbf{y}_i|C_j)$ on training data
2) Use this to identify most likely C_j for each document i in test set

$$C_i \quad = \quad \arg\max_j \hat{p}(C_j)\hat{p}(\mathbf{y}_i|C_j)$$

Simple intuition about Naive Bayes:

- Learn what documents in class j look like
- Find class j that document i is most similar to

Some R Code

```
library(e1071)
dep<- c(labels, rep(NA, no.testSet))
dep<- as.factor(dep)
out<- naiveBayes(dep∼., as.data.frame(tdm))
predicts<- predict(out, as.data.frame(tdm[-training.set,]))
```

Assessing Models (Elements of Statistical Learning)

- Model Selection: tuning parameters to select final model (next week's discussion)
- Model assessment : after selecting model, estimating error in classification

Comparing Training and Validation Set

Text classification and model assessment

- Replicate classification exercise with validation set
- General principle of classification/prediction
- Compare supervised learning labels to hand labels

Confusion matrix

Comparing Training and Validation Set

Representation of Test Statistics from Week 3 (along with some new ones)

	Actual Label	
Classification (algorithm)	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

Comparing Training and Validation Set

Representation of Test Statistics from Week 3 (along with some new ones)

	Actual Label	
Classification (algorithm)	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

$$\text{Accuracy} = \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}}$$

Comparing Training and Validation Set

Representation of Test Statistics from Week 3 (along with some new ones)

	Actual Label	
Classification (algorithm)	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

$$\text{Accuracy} = \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}}$$

$$\text{Precision}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Liberal}}$$

Comparing Training and Validation Set

Representation of Test Statistics from Week 3 (along with some new ones)

	Actual Label	
Classification (algorithm)	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

$$\text{Accuracy} = \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}}$$

$$\text{Precision}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Liberal}}$$

$$\text{Recall}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Conservative}}$$

Comparing Training and Validation Set

Representation of Test Statistics from Week 3 (along with some new ones)

	Actual Label	
Classification (algorithm)	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

$$\text{Accuracy} = \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}}$$

$$\text{Precision}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Liberal}}$$

$$\text{Recall}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Conservative}}$$

$$F_{\text{Liberal}} = \frac{2\text{Precision}_{\text{Liberal}}\text{Recall}_{\text{Liberal}}}{\text{Precision}_{\text{Liberal}} + \text{Recall}_{\text{Liberal}}}$$

Comparing Training and Validation Set

Representation of Test Statistics from Week 3 (along with some new ones)

	Actual Label	
Classification (algorithm)	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

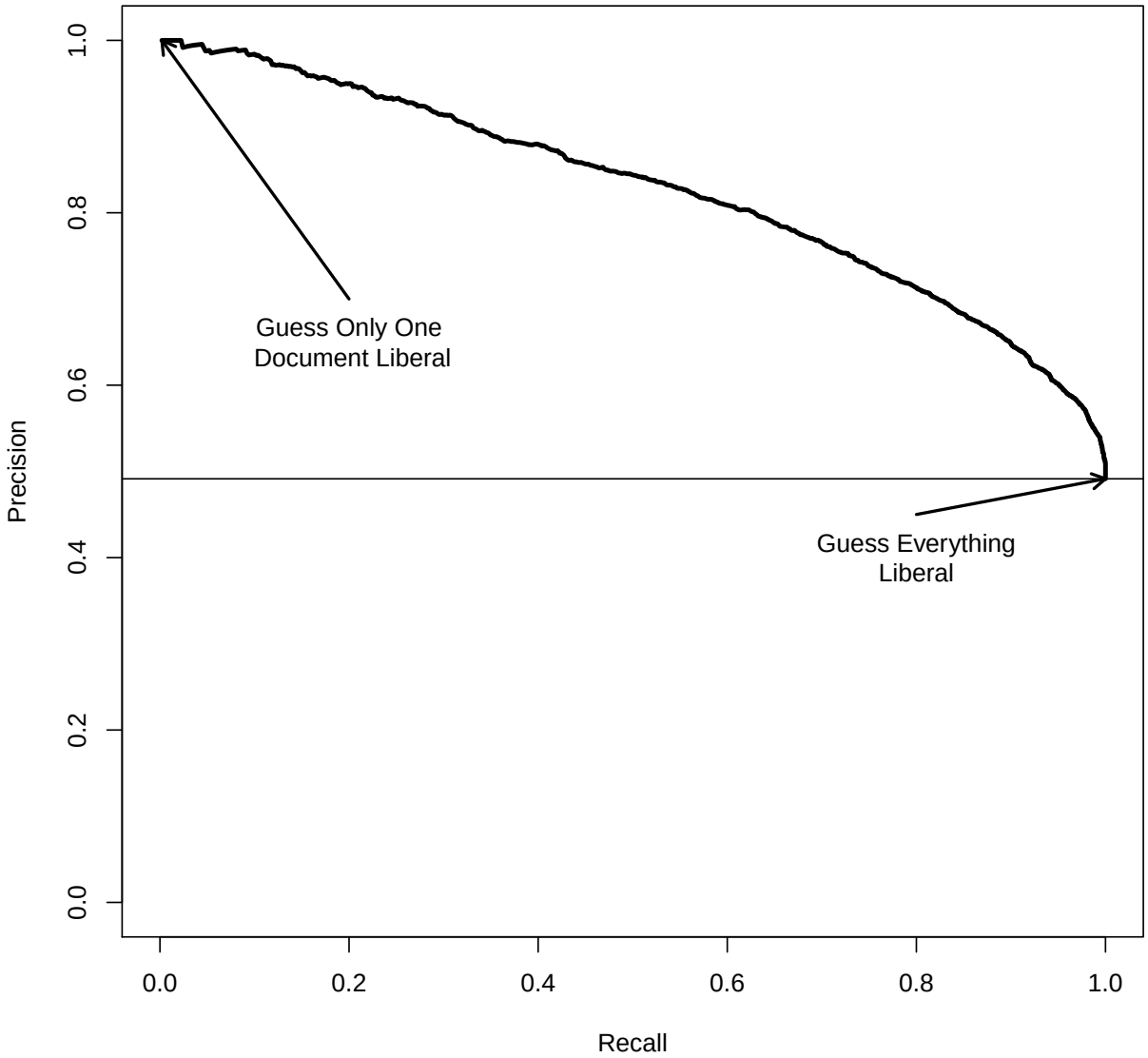
$$\text{Accuracy} = \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}}$$

$$\text{Precision}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Liberal}}$$

$$\text{Recall}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Conservative}}$$

$$F_{\text{Liberal}} = \frac{2\text{Precision}_{\text{Liberal}}\text{Recall}_{\text{Liberal}}}{\text{Precision}_{\text{Liberal}} + \text{Recall}_{\text{Liberal}}}$$

Precision Recall Tradeoff



ROC Curve

Inspires: ROC as a measure of model performance

$$\text{Recall}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Conservative}}$$

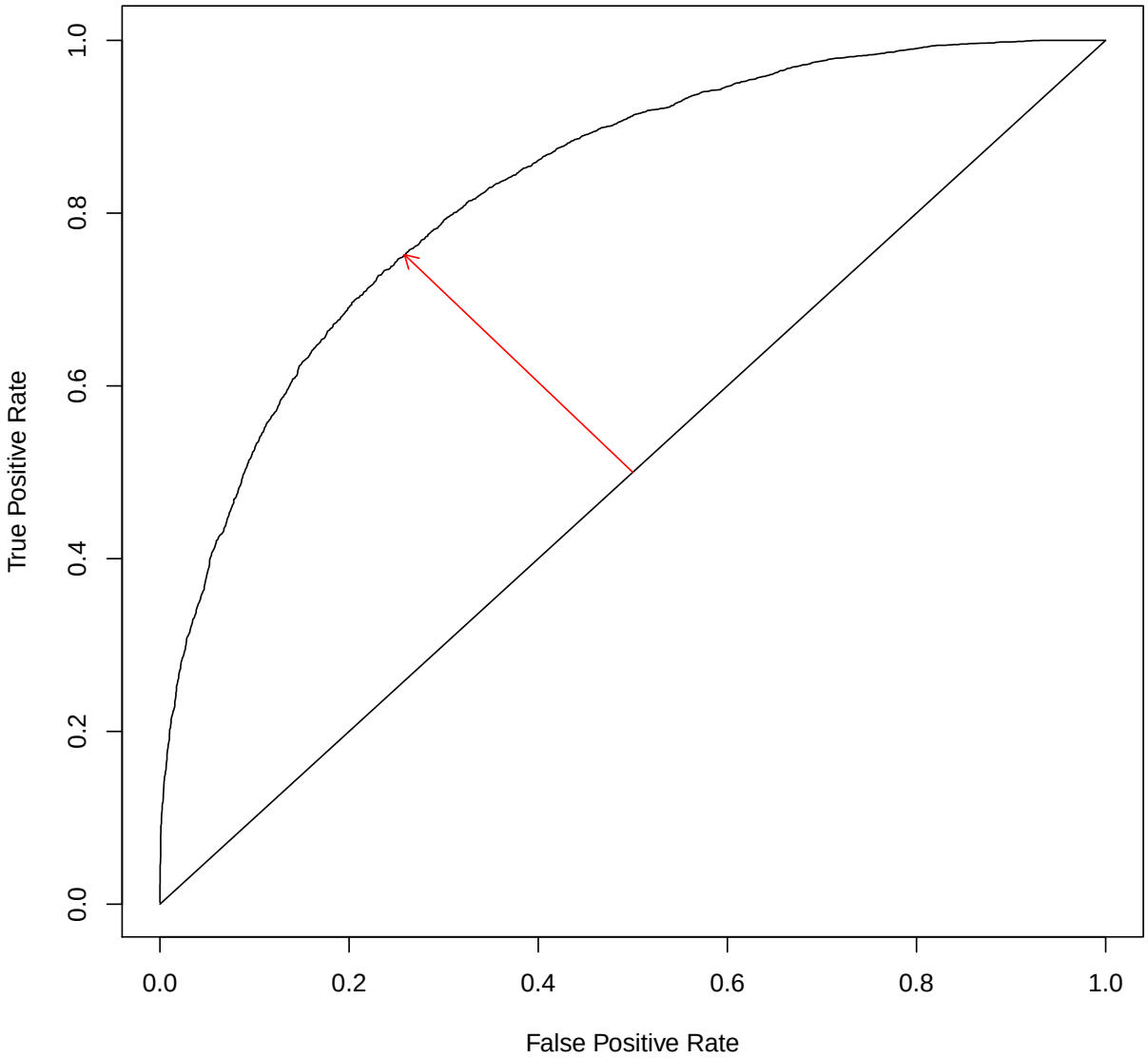
$$\text{Recall}_{\text{Conservative}} = \frac{\text{True Conservative}}{\text{True Conservative} + \text{False Liberal}}$$

Tension:

- Everything liberal: $\text{Recall}_{\text{Liberal}} = 1$; $\text{Recall}_{\text{Conservative}} = 0$
- Everything conservative: $\text{Recall}_{\text{Liberal}} = 0$; $\text{Recall}_{\text{Conservative}} = 1$

Characterize Tradeoff:
Plot True Positive Rate $\text{Recall}_{\text{Liberal}}$
False Positive Rate (1 - $\text{Recall}_{\text{Conservative}}$)

Precision/Recall Tradeoff



Simple Classification Example

Analyzing house press releases

Hand Code: 1,000 press releases

- Advertising
- Credit Claiming
- Position Taking

Divide 1,000 press releases into two sets

- 500: Training set
- 500: Test set

Initial exploration: provides baseline measurement at classifier performances

Improve: through improving model fit

Example from Ongoing Work

	Actual Label		
Classification (Naive Bayes)	Position Taking	Advertising	Credit Claim.
Position Taking	10	0	0
Advertising	2	40	2
Credit Claiming	80	60	306

$$\begin{aligned}
\text{Accuracy} &= \frac{10+40+306}{500} = 0.71 \\
\text{Precision}_{PT} &= \frac{10}{10} = 1 \\
\text{Recall}_{PT} &= \frac{10}{10+2+80} = 0.11 \\
\text{Precision}_{AD} &= \frac{40}{40+2+2} = 0.91 \\
\text{Recall}_{AD} &= \frac{40}{40+60} = 0.4 \\
\text{Precision}_{Credit} &= \frac{306}{306+80+60} = 0.67 \\
\text{Recall}_{Credit} &= \frac{306}{306+2} = 0.99
\end{aligned}$$

Fit Statistics in R

`RWeka` library provides **Amazing** functionality.
We'll have more to say on how to install, use this next week!

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes (and next week, SVM): focused on individual document classification.

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes (and next week, SVM): focused on individual document classification.
But what if we're focused on proportions only?

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes (and next week, SVM): focused on individual document classification.
But what if we're focused on proportions only?
Hopkins and King (2010): method for characterizing distribution of classes

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes (and next week, SVM): focused on individual document classification.
But what if we're focused on proportions only?
Hopkins and King (2010): method for characterizing distribution of classes
Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes (and next week, SVM): focused on individual document classification.
But what if we're focused on proportions only?
Hopkins and King (2010): method for characterizing distribution of classes
Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes (and next week, SVM): focused on individual document classification.
But what if we're focused on proportions only?
Hopkins and King (2010): method for characterizing distribution of classes
Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies
- Hopkins and King (2010): extend the method to text documents

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes (and next week, SVM): focused on individual document classification.
But what if we're focused on proportions only?
Hopkins and King (2010): method for characterizing distribution of classes
Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies
- Hopkins and King (2010): extend the method to text documents

Basic intuition:

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes (and next week, SVM): focused on individual document classification.
But what if we're focused on proportions only?
Hopkins and King (2010): method for characterizing distribution of classes
Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies
- Hopkins and King (2010): extend the method to text documents

Basic intuition:

- Examine joint distribution of characteristics (without making Naive Bayes like assumption)
- Focus on distributions (only) makes this analysis possible

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(Mx1)$ vector]

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(Mx1)$ vector]

$$\mathbf{y}_i \quad = \quad (1, 0, 0, 1, \ldots, 0)$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(Mx1)$ vector]

$$\mathbf{y}_i \quad = \quad (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\mathbf{y}_i$?

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(Mx1)$ vector]

$$\mathbf{y}_i \quad = \quad (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\mathbf{y}_i$?

- 2^M possible vectors

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(Mx1)$ vector]

$$\mathbf{y}_i \quad = \quad (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\mathbf{y}_i$?

- 2^M possible vectors

Define:

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(Mx1)$ vector]

$$\mathbf{y}_i \quad = \quad (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\mathbf{y}_i$?

- 2^M possible vectors

Define:

$$P(\mathbf{y}) \quad = \quad \text{probability of observing } \mathbf{y}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(Mx1)$ vector]

$$\mathbf{y}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\mathbf{y}_i$?

- 2^M possible vectors

Define:

$$\begin{aligned} P(\mathbf{y}) &= \text{probability of observing } \mathbf{y} \\ P(\mathbf{y}|C_j) &= \text{Probability of observing } \mathbf{y} \text{ conditional on category } C_j \end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(M x 1)$ vector]

$$\mathbf{y}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\mathbf{y}_i$?

- 2^M possible vectors

Define:

$$\begin{aligned} P(\mathbf{y}) &= \text{probability of observing } \mathbf{y} \\ P(\mathbf{y}|C_j) &= \text{Probability of observing } \mathbf{y} \text{ conditional on category } C_j \\ P(\mathbf{y}|C) &= \text{Matrix collecting vectors} \end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(Mx1)$ vector]

$$\mathbf{y}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\mathbf{y}_i$?

- 2^M possible vectors

Define:

$$\begin{aligned} P(\mathbf{y}) &= \text{probability of observing } \mathbf{y} \\ P(\mathbf{y}|C_j) &= \text{Probability of observing } \mathbf{y} \text{ conditional on category } C_j \\ P(\mathbf{y}|C) &= \text{Matrix collecting vectors} \\ P(C) &= P(C_1, C_2, \ldots, C_J) \text{ target quantity of interest} \end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(Mx1)$ vector]

$$\mathbf{y}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\mathbf{y}_i$?

- 2^M possible vectors

Define:

$$\begin{aligned} P(\mathbf{y}) &= \text{probability of observing } \mathbf{y} \\ P(\mathbf{y}|C_j) &= \text{Probability of observing } \mathbf{y} \text{ conditional on category } C_j \\ P(\mathbf{y}|C) &= \text{Matrix collecting vectors} \\ P(C) &= P(C_1, C_2, \ldots, C_J) \text{ target quantity of interest} \end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

$$\underbrace{P(\mathbf{y})}_{2^M x 1} = \underbrace{P(\mathbf{y}|C)}_{2^M x J}\underbrace{P(C)}_{J x 1}$$

Matrix algebra problem to solve, for $P(C)$
Like Naive Bayes, requires two pieces to estimate
Complication $2^M >>$ no. documents
Kernel Smoothing Methods (without a formal model)

- $P(\mathbf{y})$ = estimate directly from test set
- $P(\mathbf{y}|C)$ = estimate from training set
  - Key assumption: $P(\mathbf{y}|C)$ in training set is equivalent to $P(\mathbf{y}|C)$ in test set
- If true, can perform biased sampling of documents, worry less about drift...

Algorithm Summarized

- Estimate $\hat{p}(\mathbf{y})$ from test set
- Estimate $\hat{p}(\mathbf{y}|C)$ from training set
- Use $\hat{p}(\mathbf{y})$ and $\hat{p}(\mathbf{y}|C)$ to solve for $p(C)$

Assessing Model Performance

Not classifying individual documents $\rightarrow$ different standards
Mean Square Error(ESL, Wikipedia) :

$$\mathrm{E}[(\hat{\theta} - \theta)^2] = \mathrm{var}(\hat{\theta}) + \mathrm{Bias}(\hat{\theta}, \theta)^2$$

Suppose we have true proportions $P(C)^{\text{true}}$. Then, we'll estimate Root Mean Square Error

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{J}(P(C_j)^{\text{true}} - P(C_j))}{J}}$$

$$\text{Mean Abs. Prediction Error} = |\frac{\sum_{j=1}^{J}(P(C_j)^{\text{true}} - P(C_j))}{J}|$$

Visualize: plot true and estimated proportions

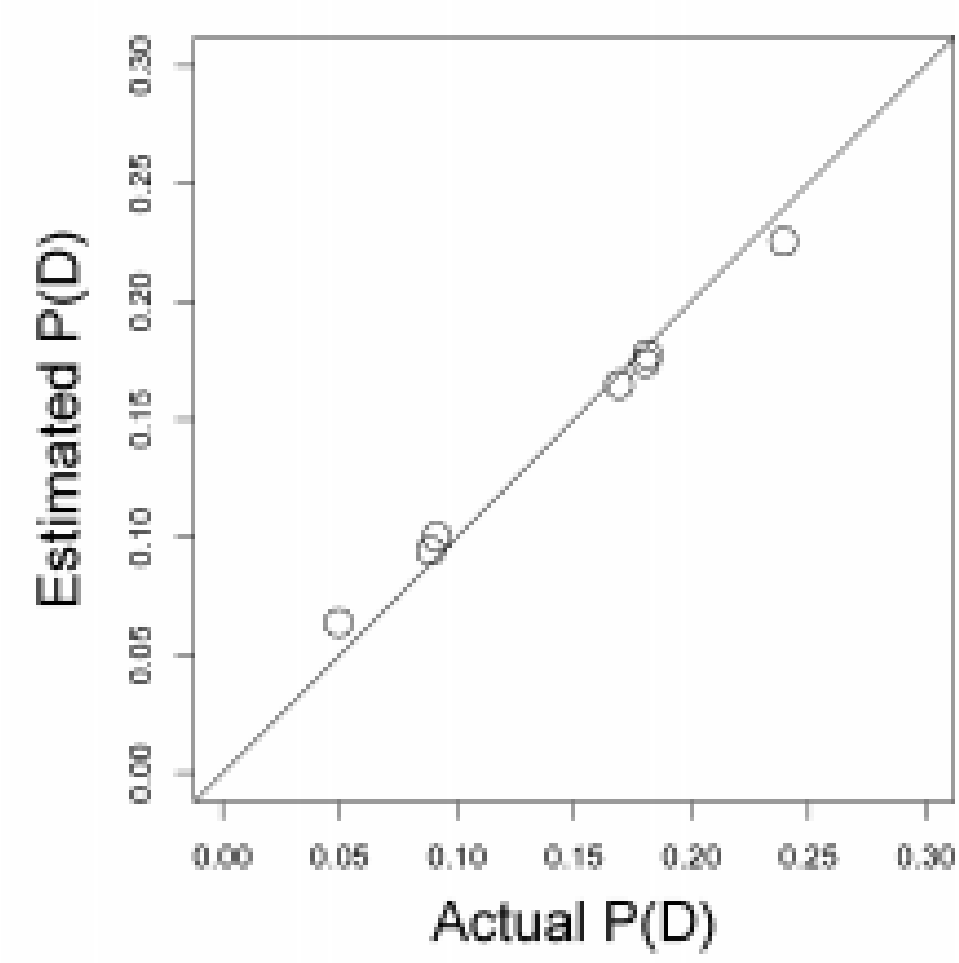

Estimated P(D)
Actual P(D)
0.00
0.05
0.10
0.15
0.20
0.25
0.30

TABLE 1 Performance of Our Nonparametric Approach and Four Support Vector Machine Analyses

	Percent of Blog Posts Correctly Classified			
	In-Sample Fit	In-Sample Cross-Validation	Out-of-Sample Prediction	Mean Absolute Proportion Error
Nonparametric	—	—	—	1.2
Linear	67.6	55.2	49.3	7.7
Radial	67.6	54.2	49.1	7.7
Polynomial	99.7	48.9	47.8	5.3
Sigmoid	15.6	15.6	18.2	23.2

Notes: Each row is the optimal choice over numerous individual runs given a specific kernel. Leaving aside the sigmoid kernel, individual classification performance in the first three columns does not correlate with mean absolute error in the document category proportions in the last column.

Using the House Press Release Data

Method	RMSE	APSE
ReadMe	0.036	0.056
NaiveBayes	0.096	0.14
SVM	0.052	0.084

Code to Run in R

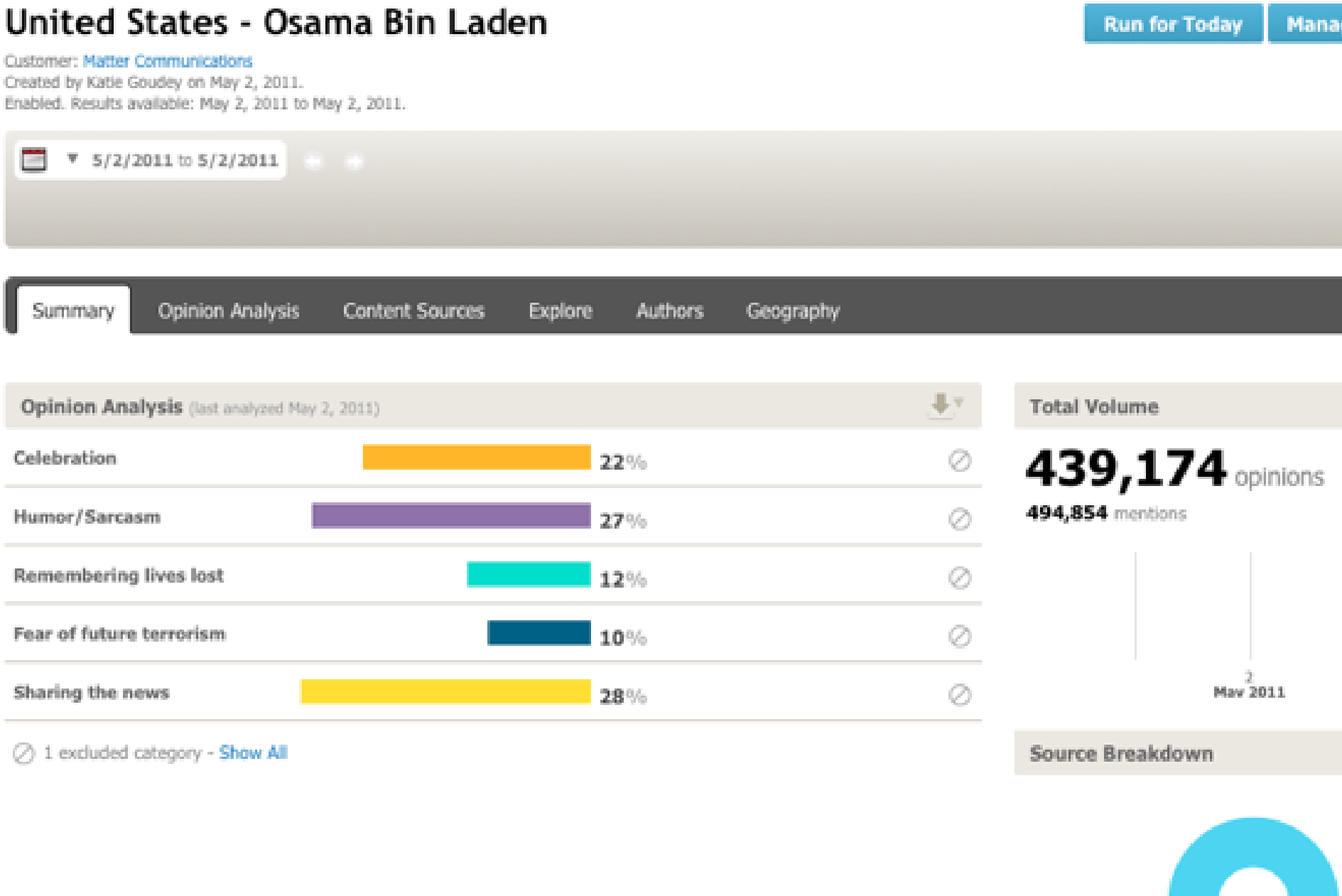
I will post code–program requires some small modifications

Control file:

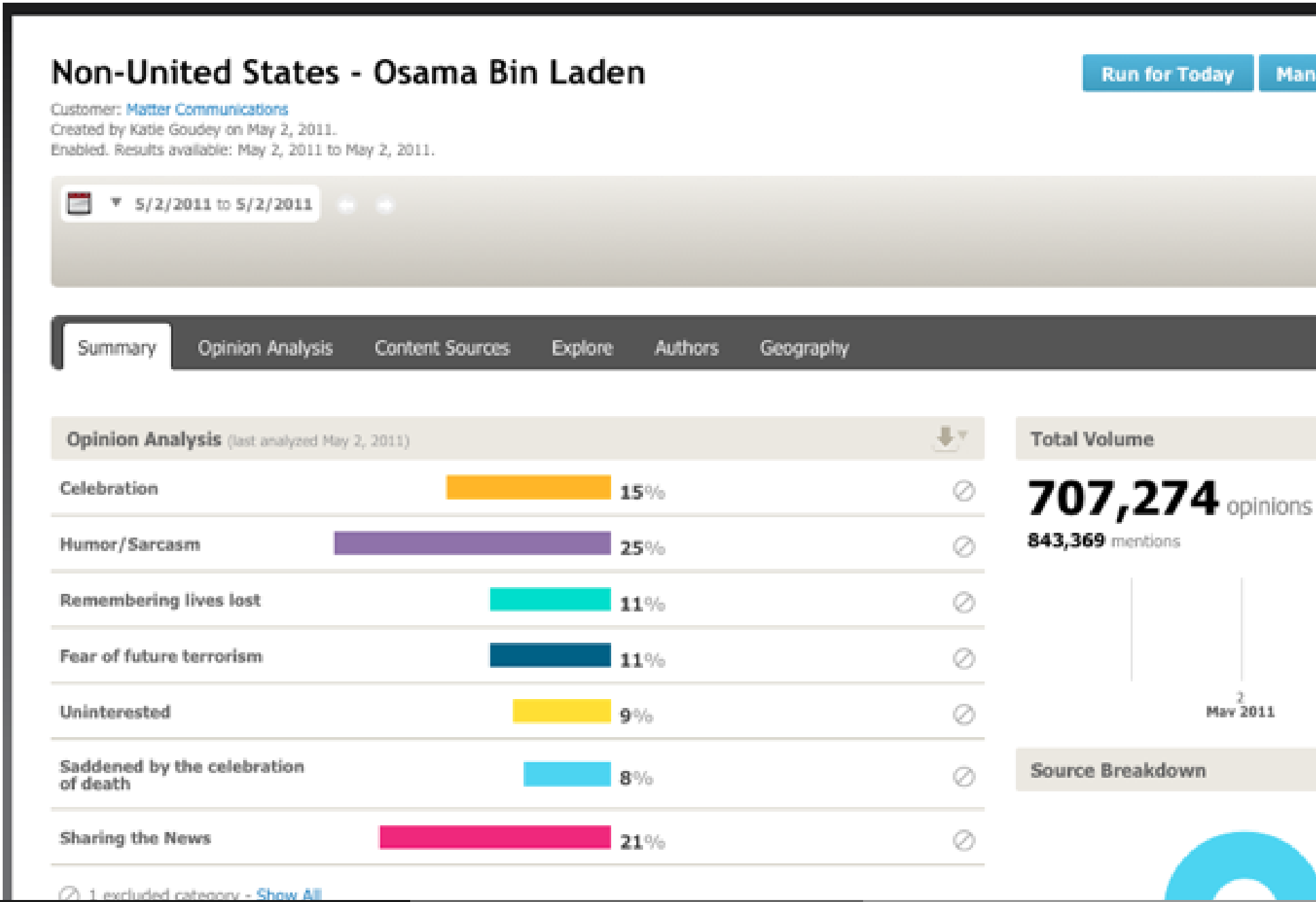
filename	truth	trainingset
20July2009LEWIS53.txt	4	1
26July2006LEWIS249.txt	2	0

```
tdm<- undergrad(control=control, fullfreq=F)
process<- preprocess(tdm)
output<- undergrad(process)
output$est.CSMF ## proportion in each category
output$true.CSMF ## if labeled for validation set (but not
used in training set)
```

Twitter and ReadMe



Twitter and ReadMe



Where we're going

Next week: cross validation to perform model selection/validation