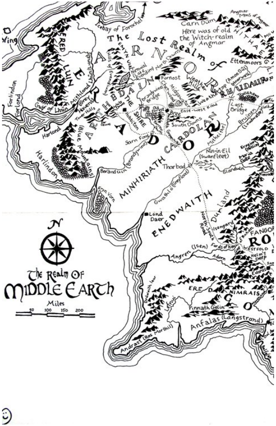
Web Scraping & Text Mining

Paulo Serôdio

Postdoctoral Researcher
School of Economics
Universitat de Barcelona

May 18, 2018

Where Are We?



We've covered the basics of document representation and characterization.

Now begin to think about documents as members of categories or classes

→ simple, fast dictionary based ways to classify/categorize

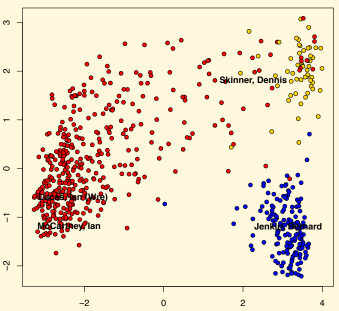
cover some 'major' dictionaries in social science

and move on to supervised learning problems.

Terminology

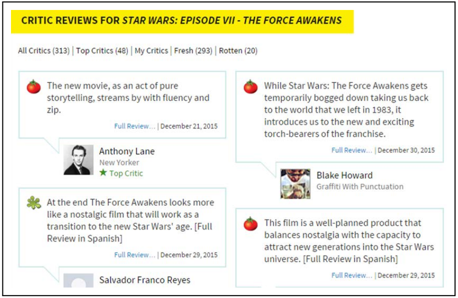
<u>Unsupervised</u> techniques: learning (hidden or latent) structure in unlabeled data.

e.g. PCA of legislators's votes: want to see how they are organized—by party? by ideology? by race?



<u>Supervised</u> techniques: learning relationship between inputs and a labeled set of outputs.

e.g. opinion mining: what makes a critic like or dislike a movie ($y \in \{0,1\}$)?



Overview: Supervised Learning

label some examples of each category

- e.g. some reviews that were positive ($y = 1$), some that were negative ($y = 0$); some statements that were liberal, some that were conservative.

train a 'machine' on these examples (e.g. logistic regression), using the features (DTM, other stuff) as the 'independent' variables.

- e.g. does the commentator use the word '`fetus`' or '`baby`' in discussing abortion law?

classify use the learned relationship to predict the outcomes of documents ($y \in \{0, 1\}$, review sentiment) not in the training set.

Overview

idea: set of pre-defined words with specific connotations that allow us to classify documents automatically, quickly and accurately.

$\rightarrow$ common in opinion mining/sentiment analysis, and in coding events or manifestos.

Often derived from supervised learning techniques

and often used in supervised learning problems, as a starting point.

so we'll cover them here in that context.

Estimating Word Discrimination

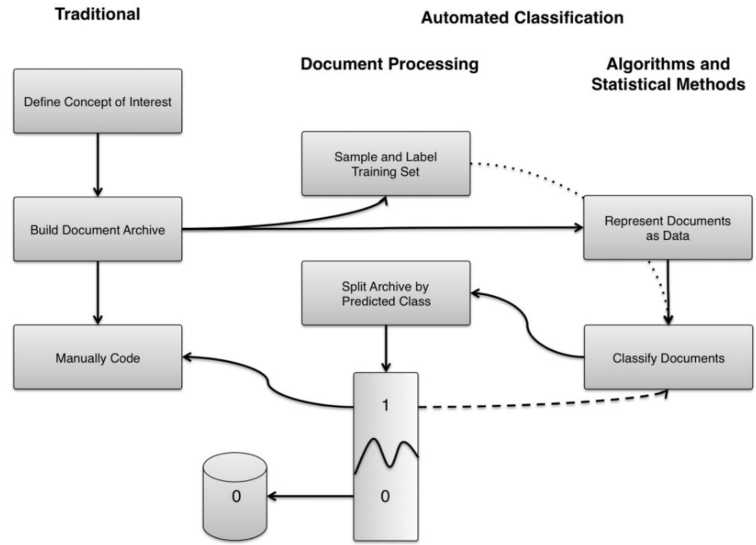


Fig. 1 The data collection process.

Estimating Word Discrimination

1) Task
   a) Classification $\rightsquigarrow$ learn word weights for dictionaries
   b) Fictitious prediction problem $\rightsquigarrow$ Identify features that discriminate between groups to learn features that are indicative of some group
2) Objective function

$$f(\boldsymbol{\theta}, \boldsymbol{X}) = f(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{Y})$$

where:

$\boldsymbol{Y}$ = Document Labels
$\boldsymbol{X}$ = Document Features
$\boldsymbol{\theta}$ = Parameters that measure words discrimination between categories

3) Optimization $\rightsquigarrow$ method specific
4) Validation $\rightsquigarrow$ depends on task
   i) Classification $\rightsquigarrow$ Accuracy, Precision, Recall
   ii) Fictitious prediction $\rightsquigarrow$ Face, convergent, discriminatory, and confound

Stylometry$\rightsquigarrow$ Who Wrote Disputed Federalist Papers?

Federalist papers $\rightsquigarrow$ Mosteller and Wallace (1963)

- Persuade citizens of New York State to adopt constitution
- Canonical texts in study of American politics
- 77 essays
  - Published from 1787-1788 in Newspapers
  - And under the name Publius, anonymously

Who Wrote the Federalist papers?

- Jay wrote essays 2, 3, 4,5, and 64
- Hamilton: wrote 43 papers
- Madison: wrote 12 papers

Disputed: Hamilton or Madison?

- Essays: 49-58, 62, and 63
- Joint Essays: 18-20

Task: identify authors of the disputed papers.

Task: Classify papers as Hamilton or Madison using dictionary methods

Setting up the Analysis

Training $\rightsquigarrow$ papers Hamilton, Madison are known to have authored
Test $\rightsquigarrow$ unlabeled papers
Preprocessing:

- Hamilton/Madison both discuss similar issues
- Differ in extent they use stop words
- Focus analysis on the stop words

Setting up the Analysis

- $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_N) =$ (Hamilton, Hamilton, Madison, ..., Hamilton)
  $N \times 1$ matrix with author labels
- Define the number of words in federalist paper i as num_i

$$\boldsymbol{X} = \begin{pmatrix} \frac{1}{\text{num}_1} & \frac{2}{\text{num}_1} & \frac{0}{\text{num}_1} & \cdots & \frac{3}{\text{num}_1} \\ \frac{0}{\text{num}_2} & \frac{1}{\text{num}_2} & \frac{0}{\text{num}_2} & \cdots & \frac{0}{\text{num}_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{0}{\text{num}_N} & \frac{0}{\text{num}_N} & \frac{1}{\text{num}_N} & \cdots & \frac{0}{\text{num}_N} \end{pmatrix}$$

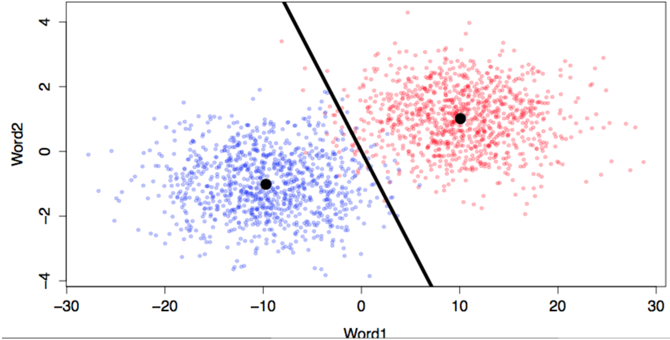
  $N \times J$ counting stop word usage rate
- $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_J)$
  Word weights.

Objective Function

Heuristically: find $\boldsymbol{\theta}^* = (\theta_1^*, \theta_2^*, \ldots, \theta_J^*)$ used to create score

$$p_i = \sum_{j=1}^{J} \theta_j^* X_{ij}$$

that maximally discriminates between categories



Objective Function

Define:

$$
\begin{aligned}
\boldsymbol{\mu}_{\text{Madison}} &= \frac{1}{N_{\text{Madison}}} \sum_{i=1}^{N} I(Y_i = \text{Madison}) \boldsymbol{X}_i \\
\boldsymbol{\mu}_{\text{Hamilton}} &= \frac{1}{N_{\text{Hamilton}}} \sum_{i=1}^{N} I(Y_i = \text{Hamilton}) \boldsymbol{X}_i
\end{aligned}
$$

Objective Function

We can then define functions that describe the “projected” mean and variance for each author

$$
\begin{aligned}
g(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{Y}, \text{Madison}) &= \frac{1}{N_{\text{Madison}}} \sum_{i=1}^{N} I(Y_i = \text{Madison})\boldsymbol{\theta}^{'}\boldsymbol{X}_i = \boldsymbol{\theta}^{'}\boldsymbol{\mu}_{\text{Madison}} \\
g(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{Y}, \text{Hamilton}) &= \frac{1}{N_{\text{Hamilton}}} \sum_{i=1}^{N} I(Y_i = \text{Hamilton})\boldsymbol{\theta}^{'}\boldsymbol{X}_i = \boldsymbol{\theta}^{'}\boldsymbol{\mu}_{\text{Hamilton}} \\
s(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{Y}, \text{Madison}) &= \sum_{i=1}^{N} I(Y_i = \text{Madison})(\boldsymbol{\theta}^{'}\boldsymbol{X}_i - \boldsymbol{\theta}^{'}\boldsymbol{\mu}_{\text{Madison}})^2 \\
s(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{Y}, \text{Hamilton}) &= \sum_{i=1}^{N} I(Y_i = \text{Hamilton})(\boldsymbol{\theta}^{'}\boldsymbol{X}_i - \boldsymbol{\theta}^{'}\boldsymbol{\mu}_{\text{Hamilton}})^2
\end{aligned}
$$

Objective Function $\rightsquigarrow$ Optimization

$$
\begin{aligned}
f(\theta, \boldsymbol{X}, \boldsymbol{Y}) &= \frac{(g(\theta, \boldsymbol{X}, \boldsymbol{Y}, \text{Hamilton}) - g(\theta, \boldsymbol{X}, \boldsymbol{Y}, \text{Madison}))^2}{s(\theta, \boldsymbol{X}, \boldsymbol{Y}, \text{Hamilton}) + s(\theta, \boldsymbol{X}, \boldsymbol{Y}, \text{Madison})} \\
&= \frac{\left(\theta^{'}(\mu_{\text{Hamilton}} - \mu_{\text{Madison}})\right)^2}{\text{Scatter}_{\text{Hamilton}} + \text{Scatter}_{\text{Madison}}}
\end{aligned}
$$

Optimization $\rightsquigarrow$ find θ^* to maximize $f(\theta, \boldsymbol{X}, \boldsymbol{Y})$, assuming independence across dimensions.

(Fisher's) Linear Discriminant Analysis

Optimization⇝ Word Weights

For each word j, construct weight θ_j^*,

$$
\begin{aligned}
\mu_{j,\text{Hamilton}} &= \frac{\sum_{i=1}^{N} I(Y_i = \text{Hamilton})X_{ij}}{\sum_{j=1}^{J}\sum_{i=1}^{N} I(Y_i = \text{Hamilton})X_{ij}} \\
\mu_{j,\text{Madison}} &= \frac{\sum_{i=1}^{N} I(Y_i = \text{Madison})X_{ij}}{\sum_{j=1}^{J}\sum_{i=1}^{N} I(Y_i = \text{Madison})X_{ij}} \\
\sigma^2_{j,\text{Hamilton}} &= \text{Var}(X_{i,j}|\text{Hamilton}) \\
\sigma^2_{j,\text{Madison}} &= \text{Var}(X_{i,j}|\text{Madison})
\end{aligned}
$$

We can then generate weight θ_j^* as

$$
\theta_j^* = \frac{\mu_{j,\text{Hamilton}} - \mu_{j,\text{Madison}}}{\sigma^2_{j,\text{Hamilton}} + \sigma^2_{j,\text{Madison}}}
$$

Optimization$\rightsquigarrow$ Trimming the Dictionary

- Trimming weights: Focus on discriminating words (very simple regularization)
- Cut off: For all $|\theta_j^*| < 0.025$ set $\theta_j^* = 0$.

Classification$\rightsquigarrow$ Determining Authorship

For each disputed document i, compute discrimination statistic

$$p_i = \sum_{j=1}^{J} \theta_j^* X_{ij}$$

$p_i \rightsquigarrow$ classification (linear discriminator)

- Above midpoint in training set $\rightarrow$ Hamilton text
- Below midpoint in training set $\rightarrow$ Madison text

Findings: Madison is the author of the disputed federalist papers.

Inferring Separating Words

Classification$\rightsquigarrow$ Custom Dictionaries

Inferring Separating Words

Classification $\rightsquigarrow$ Custom Dictionaries

- Stylometry $\rightsquigarrow$ Classify Authors

Inferring Separating Words

Classification$\rightsquigarrow$ Custom Dictionaries

- Stylometry$\rightsquigarrow$ Classify Authors
- Dictionary based classification $\rightsquigarrow$ Gentzkow and Shapiro (2010) and measures of media slant

Inferring Separating Words

Classification$\rightsquigarrow$ Custom Dictionaries

- Stylometry$\rightsquigarrow$ Classify Authors
- Dictionary based classification $\rightsquigarrow$ Gentzkow and Shapiro (2010) and measures of media slant
- Dictionary based classification$\rightsquigarrow$ Customized to particular setting

Inferring Separating Words

Classification $\rightsquigarrow$ Custom Dictionaries

- Stylometry $\rightsquigarrow$ Classify Authors
- Dictionary based classification $\rightsquigarrow$ Gentzkow and Shapiro (2010) and measures of media slant
- Dictionary based classification $\rightsquigarrow$ Customized to particular setting

Fictitious Prediction Problem $\rightsquigarrow$ Infer words that are indicative of some class/group

Inferring Separating Words

Classification $\rightsquigarrow$ Custom Dictionaries

- Stylometry $\rightsquigarrow$ Classify Authors
- Dictionary based classification $\rightsquigarrow$ Gentzkow and Shapiro (2010) and measures of media slant
- Dictionary based classification $\rightsquigarrow$ Customized to particular setting

Fictitious Prediction Problem $\rightsquigarrow$ Infer words that are indicative of some class/group

- Difference in Republican, Democratic language $\rightsquigarrow$ Partisan words

Inferring Separating Words

Classification$\rightsquigarrow$ Custom Dictionaries

- Stylometry$\rightsquigarrow$ Classify Authors
- Dictionary based classification $\rightsquigarrow$ Gentzkow and Shapiro (2010) and measures of media slant
- Dictionary based classification$\rightsquigarrow$ Customized to particular setting

Fictitious Prediction Problem $\rightsquigarrow$ Infer words that are indicative of some class/group

- Difference in Republican, Democratic language $\rightsquigarrow$ Partisan words
- Difference in Liberal, Conservative language $\rightsquigarrow$ Ideological Language

Inferring Separating Words

Classification$\rightsquigarrow$ Custom Dictionaries

- Stylometry$\rightsquigarrow$ Classify Authors
- Dictionary based classification $\rightsquigarrow$ Gentzkow and Shapiro (2010) and measures of media slant
- Dictionary based classification$\rightsquigarrow$ Customized to particular setting

Fictitious Prediction Problem $\rightsquigarrow$ Infer words that are indicative of some class/group

- Difference in Republican, Democratic language $\rightsquigarrow$ Partisan words
- Difference in Liberal, Conservative language $\rightsquigarrow$ Ideological Language
- Difference in Secret/Not Secret Language $\rightsquigarrow$ Secretive Language (Gill and Spirling 2014)

Inferring Separating Words

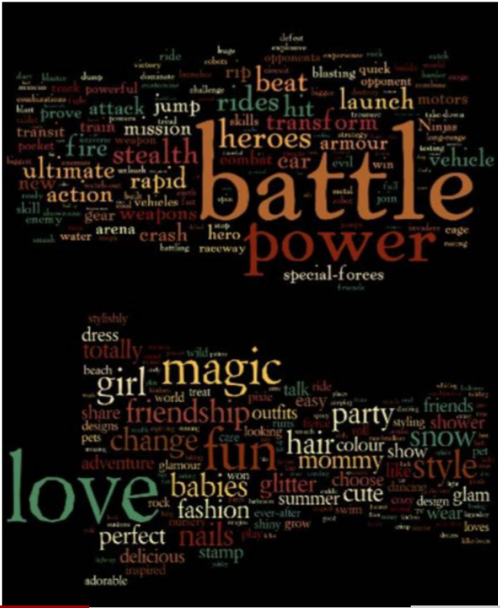
Classification $\rightsquigarrow$ Custom Dictionaries

- Stylometry $\rightsquigarrow$ Classify Authors
- Dictionary based classification $\rightsquigarrow$ Gentzkow and Shapiro (2010) and measures of media slant
- Dictionary based classification $\rightsquigarrow$ Customized to particular setting

Fictitious Prediction Problem $\rightsquigarrow$ Infer words that are indicative of some class/group

- Difference in Republican, Democratic language $\rightsquigarrow$ Partisan words
- Difference in Liberal, Conservative language $\rightsquigarrow$ Ideological Language
- Difference in Secret/Not Secret Language $\rightsquigarrow$ Secretive Language (Gill and Spirling 2014)
- Difference in Toy advertising

Inferring Separating Words



Inferring Separating Words

Classification$\rightsquigarrow$ Custom Dictionaries

- Stylometry$\rightsquigarrow$ Classify Authors
- Dictionary based classification $\rightsquigarrow$ Gentzkow and Shapiro (2010) and measures of media slant
- Dictionary based classification$\rightsquigarrow$ Customized to particular setting

Fictitious Prediction Problem $\rightsquigarrow$ Infer words that are indicative of some class/group

- Difference in Republican, Democratic language $\rightsquigarrow$ Partisan words
- Difference in Liberal, Conservative language $\rightsquigarrow$ Ideological Language
- Difference in Secret/Not Secret Language $\rightsquigarrow$ Secretive Language (Gill and Spirling 2014)
- Difference in Toy advertising
- Difference in Language across groups$\rightsquigarrow$ Labeling output from Clustering/Topic Models

Inferring Separating Words

Classification $\rightsquigarrow$ Custom Dictionaries

- Stylometry $\rightsquigarrow$ Classify Authors
- Dictionary based classification $\rightsquigarrow$ Gentzkow and Shapiro (2010) and measures of media slant
- Dictionary based classification $\rightsquigarrow$ Customized to particular setting

Fictitious Prediction Problem $\rightsquigarrow$ Infer words that are indicative of some class/group

- Difference in Republican, Democratic language $\rightsquigarrow$ Partisan words
- Difference in Liberal, Conservative language $\rightsquigarrow$ Ideological Language
- Difference in Secret/Not Secret Language $\rightsquigarrow$ Secretive Language (Gill and Spirling 2014)
- Difference in Toy advertising
- Difference in Language across groups $\rightsquigarrow$ Labeling output from Clustering/Topic Models

Vague and Difficult to derive before hand

Congressional Language Across Sources

Congressional Language Across Sources

Congressional Press Releases and Floor Speeches

Congressional Language Across Sources

Congressional Press Releases and Floor Speeches

- Collected 64,033 press releases

Congressional Language Across Sources

Congressional Press Releases and Floor Speeches

- Collected 64,033 press releases
- Problem: are they distinct from floor statements (approx. 52,000 during same time)?

Congressional Language Across Sources

Congressional Press Releases and Floor Speeches

- Collected 64,033 press releases
- Problem: are they distinct from floor statements (approx. 52,000 during same time)?
    - Yes: press releases have different purposes, targets, and need not relate to official business

Congressional Language Across Sources

Congressional Press Releases and Floor Speeches

- Collected 64,033 press releases
- Problem: are they distinct from floor statements (approx. 52,000 during same time)?
    - Yes: press releases have different purposes, targets, and need not relate to official business
    - No: press releases are just reactive to floor activity, will follow floor statements

Congressional Language Across Sources

Congressional Press Releases and Floor Speeches

- Collected 64,033 press releases
- Problem: are they distinct from floor statements (approx. 52,000 during same time)?
  - Yes: press releases have different purposes, targets, and need not relate to official business
  - No: press releases are just reactive to floor activity, will follow floor statements
- Deeper question: what does it mean for two text collections to be different?

Congressional Language Across Sources

Congressional Press Releases and Floor Speeches

- Collected 64,033 press releases
- Problem: are they distinct from floor statements (approx. 52,000 during same time)?
  - Yes: press releases have different purposes, targets, and need not relate to official business
  - No: press releases are just reactive to floor activity, will follow floor statements
- Deeper question: what does it mean for two text collections to be different?
- One Answer: texts used for different purposes

Congressional Language Across Sources

Congressional Press Releases and Floor Speeches

- Collected 64,033 press releases
- Problem: are they distinct from floor statements (approx. 52,000 during same time)?
  - Yes: press releases have different purposes, targets, and need not relate to official business
  - No: press releases are just reactive to floor activity, will follow floor statements
- Deeper question: what does it mean for two text collections to be different?
- One Answer: texts used for different purposes
- Partial answer: identify words that distinguish press releases and floor speeches

A Method for Identifying Distinguishing Words

A Method for Identifying Distinguishing Words

Mutual Information

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
    - Randomly sample a press release

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
    - Randomly sample a press release
    - Guess press release/floor statement

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
  - Randomly sample a press release
  - Guess press release/floor statement
  - Uncertainty about guess

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
  - Randomly sample a press release
  - Guess press release/floor statement
  - Uncertainty about guess
    - Maximum: No. press releases = No. floor statements

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
  - Randomly sample a press release
  - Guess press release/floor statement
  - Uncertainty about guess
    - Maximum: No. press releases = No. floor statements
    - Minimum : All documents in one category

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
    - Randomly sample a press release
    - Guess press release/floor statement
    - Uncertainty about guess
        - Maximum: No. press releases = No. floor statements
        - Minimum : All documents in one category
- Conditional uncertainty (X_j) (conditional entropy)

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
  - Randomly sample a press release
  - Guess press release/floor statement
  - Uncertainty about guess
    - Maximum: No. press releases = No. floor statements
    - Minimum : All documents in one category
- Conditional uncertainty (X_j) (conditional entropy)
  - Condition on presence of word X_j

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
  - Randomly sample a press release
  - Guess press release/floor statement
  - Uncertainty about guess
    - Maximum: No. press releases = No. floor statements
    - Minimum : All documents in one category
- Conditional uncertainty (X_j) (conditional entropy)
  - Condition on presence of word X_j
  - Randomly sample a press release

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
    - Randomly sample a press release
    - Guess press release/floor statement
    - Uncertainty about guess
        - Maximum: No. press releases = No. floor statements
        - Minimum : All documents in one category
- Conditional uncertainty (X_j) (conditional entropy)
    - Condition on presence of word X_j
    - Randomly sample a press release
    - Guess press release/floor statement

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
  - Randomly sample a press release
  - Guess press release/floor statement
  - Uncertainty about guess
    - Maximum: No. press releases = No. floor statements
    - Minimum : All documents in one category
- Conditional uncertainty (X_j) (conditional entropy)
  - Condition on presence of word X_j
  - Randomly sample a press release
  - Guess press release/floor statement
  - Word presence reduces uncertainty

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
  - Randomly sample a press release
  - Guess press release/floor statement
  - Uncertainty about guess
    - Maximum: No. press releases = No. floor statements
    - Minimum : All documents in one category
- Conditional uncertainty (X_j) (conditional entropy)
  - Condition on presence of word X_j
  - Randomly sample a press release
  - Guess press release/floor statement
  - Word presence reduces uncertainty
    - Unrelated: Conditional uncertainty = uncertainty

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
  - Randomly sample a press release
  - Guess press release/floor statement
  - Uncertainty about guess
    - Maximum: No. press releases = No. floor statements
    - Minimum : All documents in one category
- Conditional uncertainty (X_j) (conditional entropy)
  - Condition on presence of word X_j
  - Randomly sample a press release
  - Guess press release/floor statement
  - Word presence reduces uncertainty
    - Unrelated: Conditional uncertainty = uncertainty
    - Perfect predictor: Conditional uncertainty = 0

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
    - Randomly sample a press release
    - Guess press release/floor statement
    - Uncertainty about guess
        - Maximum: No. press releases = No. floor statements
        - Minimum : All documents in one category
- Conditional uncertainty (X_j) (conditional entropy)
    - Condition on presence of word X_j
    - Randomly sample a press release
    - Guess press release/floor statement
    - Word presence reduces uncertainty
        - Unrelated: Conditional uncertainty = uncertainty
        - Perfect predictor: Conditional uncertainty = 0
- Mutual information(X_j): uncertainty - conditional uncertainty (X_j)

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
  - Randomly sample a press release
  - Guess press release/floor statement
  - Uncertainty about guess
    - Maximum: No. press releases = No. floor statements
    - Minimum : All documents in one category
- Conditional uncertainty (X_j) (conditional entropy)
  - Condition on presence of word X_j
  - Randomly sample a press release
  - Guess press release/floor statement
  - Word presence reduces uncertainty
    - Unrelated: Conditional uncertainty = uncertainty
    - Perfect predictor: Conditional uncertainty = 0
- Mutual information(X_j): uncertainty - conditional uncertainty (X_j)
  - Maximum: Uncertainty $\rightarrow$ X_j is perfect predictor

A Method for Identifying Distinguishing Words

Mutual Information

- Unconditional uncertainty (entropy):
  - Randomly sample a press release
  - Guess press release/floor statement
  - Uncertainty about guess
    - Maximum: No. press releases = No. floor statements
    - Minimum : All documents in one category
- Conditional uncertainty (X_j) (conditional entropy)
  - Condition on presence of word X_j
  - Randomly sample a press release
  - Guess press release/floor statement
  - Word presence reduces uncertainty
    - Unrelated: Conditional uncertainty = uncertainty
    - Perfect predictor: Conditional uncertainty = 0
- Mutual information(X_j): uncertainty - conditional uncertainty (X_j)
  - Maximum: Uncertainty $\rightarrow X_j$ is perfect predictor
  - Minimum: $0 \rightarrow X_j$ fails to separate speeches and floor statements

A Method for Identifying Distinguishing Words

A Method for Identifying Distinguishing Words

- Pr(Press) $\equiv$ Probability selected document press release

A Method for Identifying Distinguishing Words

- $\Pr(\text{Press}) \equiv$ Probability selected document press release
- $\Pr(\text{Speech}) \equiv$ Probability selected document speech

A Method for Identifying Distinguishing Words

- $\Pr(\text{Press}) \equiv$ Probability selected document press release
- $\Pr(\text{Speech}) \equiv$ Probability selected document speech
- Define entropy $H(\text{Doc})$

A Method for Identifying Distinguishing Words

- Pr(Press) $\equiv$ Probability selected document press release
- Pr(Speech) $\equiv$ Probability selected document speech
- Define entropy $H(\text{Doc})$

$$H(\text{Doc}) = - \sum_{t \in \{\text{Pre}, \text{Spe}\}} \Pr(t) \log_2 \Pr(t)$$

A Method for Identifying Distinguishing Words

- $\Pr(\text{Press}) \equiv$ Probability selected document press release
- $\Pr(\text{Speech}) \equiv$ Probability selected document speech
- Define entropy $H(\text{Doc})$

$$H(\text{Doc}) = - \sum_{t \in \{\text{Pre}, \text{Spe}\}} \Pr(t) \log_2 \Pr(t)$$

- $\log_2$? Encodes bits

A Method for Identifying Distinguishing Words

- $\Pr(\text{Press}) \equiv$ Probability selected document press release
- $\Pr(\text{Speech}) \equiv$ Probability selected document speech
- Define entropy $H(\text{Doc})$

$$H(\text{Doc}) = - \sum_{t \in \{\text{Pre},\text{Spe}\}} \Pr(t) \log_2 \Pr(t)$$

- $\log_2$? Encodes bits
- Maximum: $\Pr(\text{Press}) = \Pr(\text{Speech}) = 0.5$

A Method for Identifying Distinguishing Words

- $\Pr(\text{Press}) \equiv$ Probability selected document press release
- $\Pr(\text{Speech}) \equiv$ Probability selected document speech
- Define entropy $H(\text{Doc})$

$$H(\text{Doc}) = - \sum_{t \in \{\text{Pre},\text{Spe}\}} \Pr(t) \log_2 \Pr(t)$$

- $\log_2$? Encodes bits
- Maximum: $\Pr(\text{Press}) = \Pr(\text{Speech}) = 0.5$
- Minimum: $\Pr(\text{Press}) \rightarrow 0$ (or $\Pr(\text{Press}) \rightarrow 1$)

A Method for Identifying Distinguishing Words

A Method for Identifying Distinguishing Words

- Consider presence/absence of word X_j

A Method for Identifying Distinguishing Words

- Consider presence/absence of word X_j
- Define conditional entropy $H(\text{Doc}|X_j)$

A Method for Identifying Distinguishing Words

- Consider presence/absence of word X_j
- Define conditional entropy $H(\text{Doc}|X_j)$

$$H(\text{Doc}|X_j) = -\sum_{s=0}^{1} \sum_{t \in \{\text{Pre},\text{Spe}\}} \Pr(t, X_j = s) \log_2 \Pr(t|X_j = s)$$

A Method for Identifying Distinguishing Words

- Consider presence/absence of word X_j
- Define conditional entropy $H(\text{Doc}|X_j)$

$$H(\text{Doc}|X_j) = -\sum_{s=0}^{1} \sum_{t \in \{\text{Pre},\text{Spe}\}} \Pr(t, X_j = s) \log_2 \Pr(t | X_j = s)$$

- Maximum: X_j unrelated to Press Releases/Floor Speeches

A Method for Identifying Distinguishing Words

- Consider presence/absence of word X_j
- Define conditional entropy $H(\text{Doc}|X_j)$

$$H(\text{Doc}|X_j) \;\; = \;\; -\sum_{s=0}^{1} \sum_{t \in \{\text{Pre,Spe}\}} \Pr(t, X_j = s) \log_2 \Pr(t | X_j = s)$$

- Maximum: X_j unrelated to Press Releases/Floor Speeches
- Minimum: X_j is a perfect predictor of press release/floor speech

A Method for Identifying Distinguishing Words

A Method for Identifying Distinguishing Words

- Define Mutual Information(X_j) as

A Method for Identifying Distinguishing Words

- Define Mutual Information(X_j) as

$$\text{Mutual Information}(X_j) = H(\text{Doc}) - H(\text{Doc}|X_j)$$

A Method for Identifying Distinguishing Words

- Define Mutual Information(X_j) as

$$\text{Mutual Information}(X_j) = H(\text{Doc}) - H(\text{Doc}|X_j)$$

- Maximum: entropy $\Rightarrow H(\text{Doc}|X_j) = 0$

A Method for Identifying Distinguishing Words

- Define Mutual Information(X_j) as

$$\text{Mutual Information}(X_j) = H(\text{Doc}) - H(\text{Doc}|X_j)$$

- Maximum: entropy $\Rightarrow H(\text{Doc}|X_j) = 0$
- Minimum: $0 \Rightarrow H(\text{Doc}|X_j) = H(\text{Doc})$.

A Method for Identifying Distinguishing Words

- Define Mutual Information(X_j) as

$$\text{Mutual Information}(X_j) = H(\text{Doc}) - H(\text{Doc}|X_j)$$

- Maximum: entropy $\Rightarrow H(\text{Doc}|X_j) = 0$
- Minimum: $0 \Rightarrow H(\text{Doc}|X_j) = H(\text{Doc})$.

Bigger mutual information $\Rightarrow$ better discrimination

A Method for Identifying Distinguishing Words

- Define Mutual Information(X_j) as

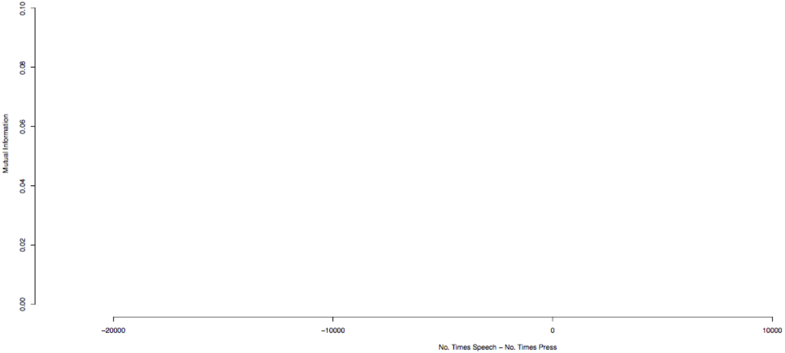
$$\text{Mutual Information}(X_j) = H(\text{Doc}) - H(\text{Doc}|X_j)$$

- Maximum: entropy $\Rightarrow H(\text{Doc}|X_j) = 0$
- Minimum: $0 \Rightarrow H(\text{Doc}|X_j) = H(\text{Doc})$.

Bigger mutual information $\Rightarrow$ better discrimination

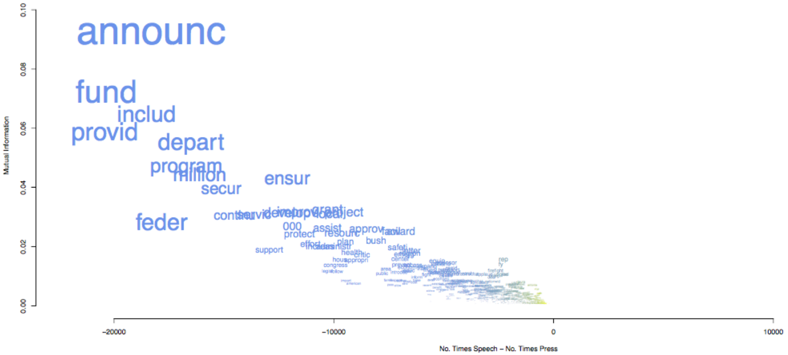
Objective function and optimization$\rightsquigarrow$ estimate probabilities that we then place in mutual information

What's Different About Press Releases



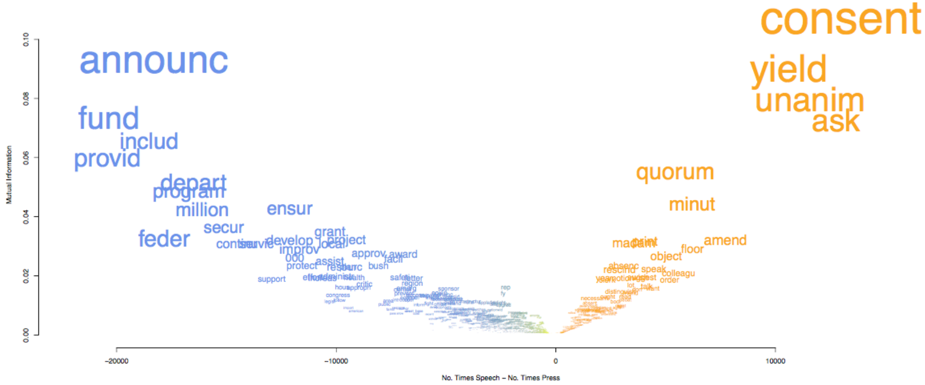
What's Different?

What's Different About Press Releases



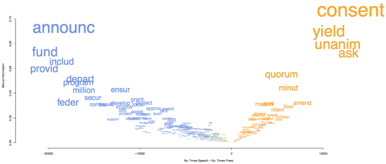
What's Different?

What's Different About Press Releases



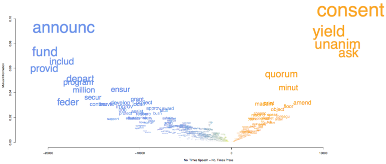
What's Different?

What's Different About Press Releases



What's Different?

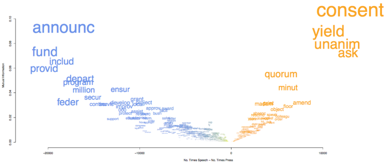
What's Different About Press Releases



What's Different?

- Press Releases: Credit Claiming

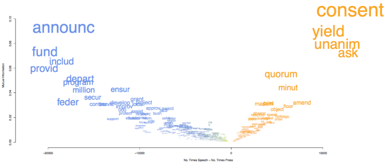
What's Different About Press Releases



What's Different?

- Press Releases: Credit Claiming
- Floor Speeches: Procedural Words

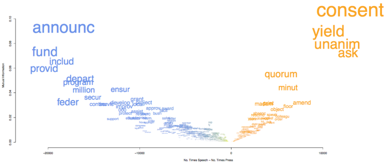
What's Different About Press Releases



What's Different?

- Press Releases: Credit Claiming
- Floor Speeches: Procedural Words
- Validate: Manual Classification

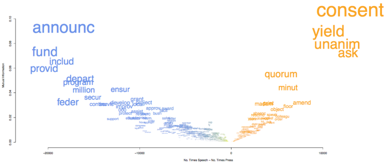
What's Different About Press Releases



What's Different?

- Press Releases: Credit Claiming
- Floor Speeches: Procedural Words
- Validate: Manual Classification
- Sample 500 Press Releases, 500 Floor Speeches

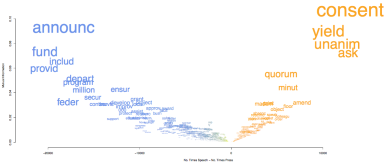
What's Different About Press Releases



What's Different?

- Press Releases: Credit Claiming
- Floor Speeches: Procedural Words
- Validate: Manual Classification
- Sample 500 Press Releases, 500 Floor Speeches
- Credit Claiming: 36% Press Releases, 4% Floor Speeches

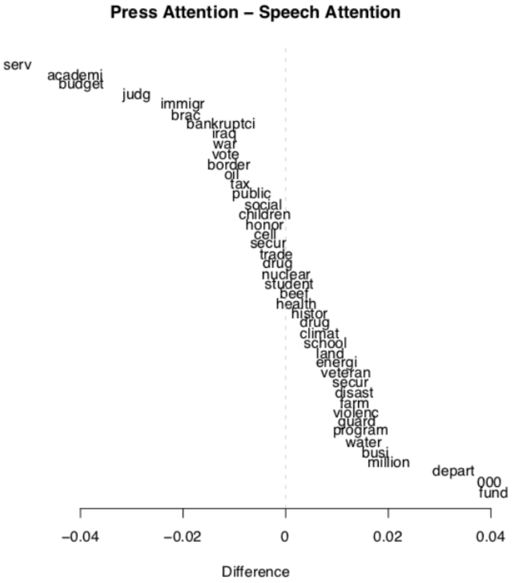
What's Different About Press Releases



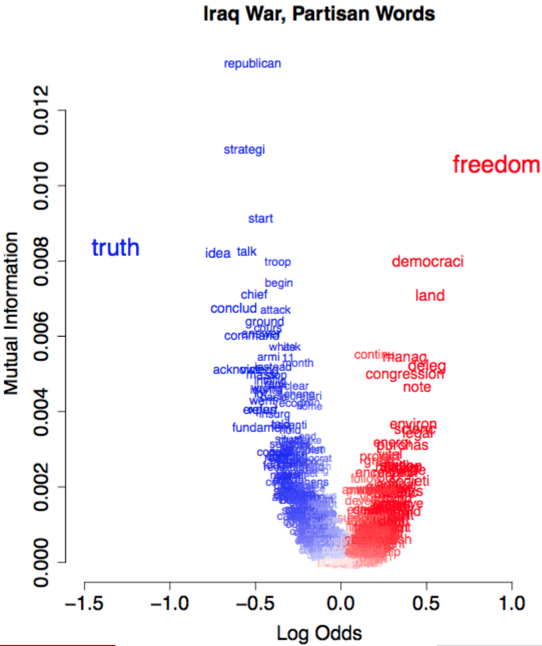
What's Different?

- Press Releases: Credit Claiming
- Floor Speeches: Procedural Words
- Validate: Manual Classification
- Sample 500 Press Releases, 500 Floor Speeches
- Credit Claiming: 36% Press Releases, 4% Floor Speeches
- Procedural: 0% Press Releases, 44% Floor Speeches

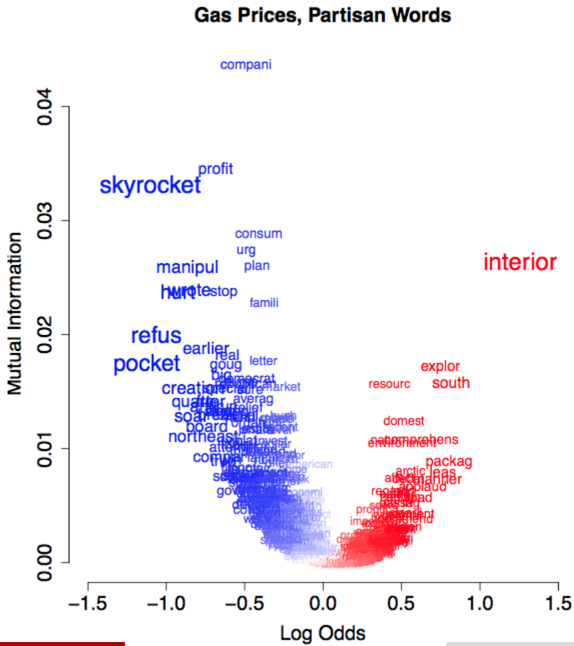
What's Different About Press Releases



Mutual Information, Standardized Log Odds



Mutual Information, Standardized Log Odds



Classification via Dictionary Methods

1) Task

Classification via Dictionary Methods

1) Task
   a) Categorize documents into predetermined categories

Classification via Dictionary Methods

1) Task
    a) Categorize documents into predetermined categories
    b) Measure documents association with predetermined categories

Classification via Dictionary Methods

1) Task
   a) Categorize documents into predetermined categories
   b) Measure documents association with predetermined categories
2) Objective function:

Classification via Dictionary Methods

1) Task
    a) Categorize documents into predetermined categories
    b) Measure documents association with predetermined categories
2) Objective function:

$$f(\boldsymbol{\theta}, \boldsymbol{X}_i) = \frac{\sum_{j=1}^{N} \theta_j X_{ij}}{\sum_{j=1}^{N} X_{ij}}$$

Classification via Dictionary Methods

1) Task
   a) Categorize documents into predetermined categories
   b) Measure documents association with predetermined categories
2) Objective function:

$$f(\boldsymbol{\theta}, \boldsymbol{X}_i) = \frac{\sum_{j=1}^{N} \theta_j X_{ij}}{\sum_{j=1}^{N} X_{ij}}$$

where:

Classification via Dictionary Methods

1) Task
   a) Categorize documents into predetermined categories
   b) Measure documents association with predetermined categories
2) Objective function:

$$f(\boldsymbol{\theta}, \boldsymbol{X}_i) \quad = \quad \frac{\sum_{j=1}^{N} \theta_j X_{ij}}{\sum_{j=1}^{N} X_{ij}}$$

where:

- $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_N)$ are word weights

Classification via Dictionary Methods

1) Task
    a) Categorize documents into predetermined categories
    b) Measure documents association with predetermined categories
2) Objective function:

$$f(\boldsymbol{\theta}, \boldsymbol{X}_i) = \frac{\sum_{j=1}^{N} \theta_j X_{ij}}{\sum_{j=1}^{N} X_{ij}}$$

where:

- $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_N)$ are word weights
- $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iN})$ count the occurrence of each corresponding word in document i

Classification via Dictionary Methods

1) Task
   a) Categorize documents into predetermined categories
   b) Measure documents association with predetermined categories
2) Objective function:

$$f(\boldsymbol{\theta}, \boldsymbol{X}_i) = \frac{\sum_{j=1}^{N} \theta_j X_{ij}}{\sum_{j=1}^{N} X_{ij}}$$

   where:
   - $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_N)$ are word weights
   - $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iN})$ count the occurrence of each corresponding word in document i
3) Optimization $\rightsquigarrow$ predetermined word list, no task specific optimization

Classification via Dictionary Methods

1) Task
   a) Categorize documents into predetermined categories
   b) Measure documents association with predetermined categories
2) Objective function:

$$f(\boldsymbol{\theta}, \boldsymbol{X}_i) = \frac{\sum_{j=1}^{N} \theta_j X_{ij}}{\sum_{j=1}^{N} X_{ij}}$$

   where:
   - $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_N)$ are word weights
   - $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iN})$ count the occurrence of each corresponding word in document i
3) Optimization$\rightsquigarrow$ predetermined word list, no task specific optimization
4) Validation (Model checking)$\rightsquigarrow$ weight (model) checking, replication of hand coding, face validity

Word Weights: Separating Classes

General Classification Goal: Place documents into categories

Word Weights: Separating Classes

General Classification Goal: Place documents into categories
How To Do Classification?

Word Weights: Separating Classes

General Classification Goal: Place documents into categories
How To Do Classification?

- Dictionaries:

Word Weights: Separating Classes

General Classification Goal: Place documents into categories
How To Do Classification?

- Dictionaries:
  - Rely on Humans $\rightsquigarrow$ humans to identify words that associate with classes

Word Weights: Separating Classes

General Classification Goal: Place documents into categories
How To Do Classification?

- Dictionaries:
  - Rely on Humans $\rightsquigarrow$ humans to identify words that associate with classes
  - Measure how well words separate (positive/negative, emotional, ...)

Word Weights: Separating Classes

General Classification Goal: Place documents into categories
How To Do Classification?

- Dictionaries:
  - Rely on Humans $\rightsquigarrow$ humans to identify words that associate with classes
  - Measure how well words separate (positive/negative, emotional, ...)
- Supervised Classification Methods:

Word Weights: Separating Classes

General Classification Goal: Place documents into categories
How To Do Classification?

- Dictionaries:
  - Rely on Humans $\rightsquigarrow$ humans to identify words that associate with classes
  - Measure how well words separate (positive/negative, emotional, ...)
- Supervised Classification Methods:
  - Rely on statistical models

Word Weights: Separating Classes

General Classification Goal: Place documents into categories
How To Do Classification?

- Dictionaries:
  - Rely on Humans$\rightsquigarrow$ humans to identify words that associate with classes
  - Measure how well words separate (positive/negative, emotional, ...)
- Supervised Classification Methods:
  - Rely on statistical models
  - Given set of coded documents, statistical relationship between classes/words

Word Weights: Separating Classes

General Classification Goal: Place documents into categories
How To Do Classification?

- Dictionaries:
    - Rely on Humans $\rightsquigarrow$ humans to identify words that associate with classes
    - Measure how well words separate (positive/negative, emotional, ...)
- Supervised Classification Methods:
    - Rely on statistical models
    - Given set of coded documents, statistical relationship between classes/words
    - Statistical measures of separation

Word Weights: Separating Classes

General Classification Goal: Place documents into categories

How To Do Classification?

- Dictionaries:
  - Rely on Humans$\rightsquigarrow$ humans to identify words that associate with classes
  - Measure how well words separate (positive/negative, emotional, ...)
- Supervised Classification Methods:
  - Rely on statistical models
  - Given set of coded documents, statistical relationship between classes/words
  - Statistical measures of separation

Key point: this is the same task

Types of Classification Problems

Topic: What is this text about?

Types of Classification Problems

Topic: What is this text about?

- Policy area of legislation
  $\Rightarrow$ {Agriculture, Crime, Environment, ...}
- Campaign agendas
  $\Rightarrow$ {Abortion, Campaign, Finance, Taxing, ... }

Types of Classification Problems

Topic: What is this text about?

- Policy area of legislation
  $\Rightarrow$ {Agriculture, Crime, Environment, ...}
- Campaign agendas
  $\Rightarrow$ {Abortion, Campaign, Finance, Taxing, ... }

Sentiment: What is said in this text? [Public Opinion]

Types of Classification Problems

Topic: What is this text about?

- Policy area of legislation
  $\Rightarrow$ {Agriculture, Crime, Environment, ...}
- Campaign agendas
  $\Rightarrow$ {Abortion, Campaign, Finance, Taxing, ... }

Sentiment: What is said in this text? [Public Opinion]

- Positions on legislation
  $\Rightarrow$ { Support, Ambiguous, Oppose }
- Positions on Court Cases
  $\Rightarrow$ { Agree with Court, Disagree with Court }
- Liberal/Conservative Blog Posts
  $\Rightarrow$ { Liberal, Middle, Conservative, No Ideology Expressed }

Types of Classification Problems

Topic: What is this text about?

- Policy area of legislation
  $\Rightarrow$ {Agriculture, Crime, Environment, ...}
- Campaign agendas
  $\Rightarrow$ {Abortion, Campaign, Finance, Taxing, ... }

Sentiment: What is said in this text? [Public Opinion]

- Positions on legislation
  $\Rightarrow$ { Support, Ambiguous, Oppose }
- Positions on Court Cases
  $\Rightarrow$ { Agree with Court, Disagree with Court }
- Liberal/Conservative Blog Posts
  $\Rightarrow$ { Liberal, Middle, Conservative, No Ideology Expressed }

Style/Tone: How is it said?

Types of Classification Problems

Topic: What is this text about?

- Policy area of legislation
  $\Rightarrow$ {Agriculture, Crime, Environment, ...}
- Campaign agendas
  $\Rightarrow$ {Abortion, Campaign, Finance, Taxing, ... }

Sentiment: What is said in this text? [Public Opinion]

- Positions on legislation
  $\Rightarrow$ { Support, Ambiguous, Oppose }
- Positions on Court Cases
  $\Rightarrow$ { Agree with Court, Disagree with Court }
- Liberal/Conservative Blog Posts
  $\Rightarrow$ { Liberal, Middle, Conservative, No Ideology Expressed }

Style/Tone: How is it said?

- Taunting in floor statements
  $\Rightarrow$ { Partisan Taunt, Intra party taunt, Agency taunt, ... }
- Negative campaigning
  $\Rightarrow$ { Negative ad, Positive ad}

Pre-existing word weights $\rightsquigarrow$ Dictionaries

Pre-existing word weights⇝ Dictionaries

DICTION

DICTION is a computer-aided text analysis program for Windows® and Mac® that uses a series of dictionaries to search a passage for five semantic features—Activity, Optimism, Certainty, Realism a Commonality—as well as thirty-five sub-features. DICTION uses predefined dictionaries and can use up to thirty custom dictionaries built with words that the user has defined, such as topical or negative words, for particular research needs.

DICTION

DICTION 7, now with *Power Mode,* can read a variety of text formats and can accept a large number files within a single project. Projects containing over 1000 files are analyzed using *power analysis* for enhanced speed and reporting efficiency, with results automatically exported to .csv-formatted spreadsheet file.

Pre-existing word weights⇝ Dictionaries

DICTION

On an average computer, DICTION can process over 20,000 passages in about five minutes. DICTIO
requires 4.9 MB of memory and 38.4 MB of hard disk space.

Pre-existing word weights⇝ Dictionaries

DICTION

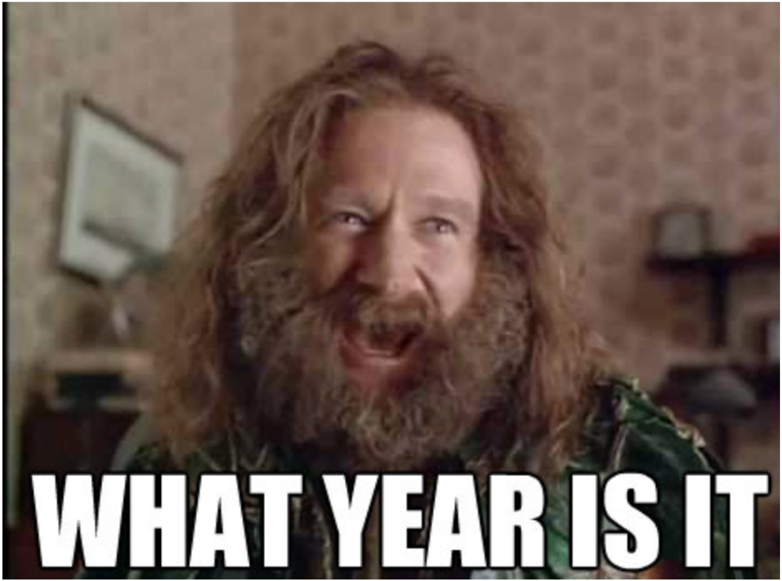
> *"provides both social scientific and humanistic understandings"*
>
> —Don Waisanen, Baruch College

Pre-existing word weights⇝ Dictionaries

`DICTION`

DICTION 7 for Mac (Educational) ($219.00)

This is the educational edition of DICTION Version 7 for Mac. You
purchase on the following page.


WHAT YEAR IS IT

Dictionary Methods

Many Dictionary Methods (like DICTION)

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary$\rightsquigarrow$ wrapped in GUI

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary $\rightsquigarrow$ wrapped in GUI
2) Basic tasks:

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary $\rightsquigarrow$ wrapped in GUI
2) Basic tasks:
    a) Count words

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary$\rightsquigarrow$ wrapped in GUI
2) Basic tasks:
    a) Count words
    b) Weighted counts of words

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary $\rightsquigarrow$ wrapped in GUI
2) Basic tasks:
   a) Count words
   b) Weighted counts of words
   c) Some graphics

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary $\rightsquigarrow$ wrapped in GUI
2) Basic tasks:
    a) Count words
    b) Weighted counts of words
    c) Some graphics
3) Pricey $\rightsquigarrow$ inexplicably

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers

Other Dictionaries

1) General Inquirer Database (http://www.wjh.harvard.edu/~inquirer/)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories$\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories$\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
     2) Judge round$\rightsquigarrow$ (a) Does the word belong? (b) What other categories might it belong to?

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories$\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
     2) Judge round$\rightsquigarrow$ (a) Does the word belong? (b) What other categories might it belong to?
   - { Positive emotion, Negative emotion }

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories $\leadsto$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
     2) Judge round $\leadsto$ (a) Does the word belong? (b) What other categories might it belong to?
   - { Positive emotion, Negative emotion }
   - 2300 words grouped into 70 classes

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories$\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
     2) Judge round$\rightsquigarrow$ (a) Does the word belong? (b) What other categories might it belong to?
   - { Positive emotion, Negative emotion }
   - 2300 words grouped into 70 classes

- Harvard-IV-4

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories$\leadsto$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
     2) Judge round$\leadsto$ (a) Does the word belong? (b) What other categories might it belong to?
   - { Positive emotion, Negative emotion }
   - 2300 words grouped into 70 classes

- Harvard-IV-4
- Affective Norms for English Words (we'll discuss this more later)

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories$\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
     2) Judge round$\rightsquigarrow$ (a) Does the word belong? (b) What other categories might it belong to?
   - { Positive emotion, Negative emotion }
   - 2300 words grouped into 70 classes

- Harvard-IV-4
- Affective Norms for English Words (we'll discuss this more later)
- The Lexicoder Sentiment Dictionary (Young and Soroka), which targeted "political

Classification with Dictionary Methods

Aim Typically we are trying to do one of two closely related things:

1. Categorize documents as belonging to a certain class (mutually exclusive? jointly exhaustive?)

   e.g. this review is 'positive', this speech is 'liberal'

2. Measure extent to which document is associated with given category

   e.g. this review is generally 'positive', but has some negative elements.

We have a pre-determined list of words, the (weighted) presence of which helps us with (1) and (2).

More Specifically

We have a set of key words, with attendant scores,

e.g. for movie reviews: '`terrible`' is scored as -1; '`fantastic`' as $+1$

$\rightarrow$ the relative rate of occurrence of these terms tells us about the overall tone or category that the document should be placed in.

i.e. for document i and words $m = 1, \ldots, M$ in the dictionary,

$$\boxed{\text{tone of document } i = \sum_{m=1}^{M} \frac{s_m w_{im}}{N_i}}$$

where s_m is the score of word m

and w_{im} is the number of occurrences of the mth dictionary word in the document i

and N_i is the total number of all dictionary words in the document.

$\rightarrow$ just add up the number of times the words appear and multiply by the score (normalizing by doc dictionary presence)

(Simple) Example: Barnes' review of *The Big Short*

Director and co-screenwriter Adam McKay (Step Brothers) bungles a great opportunity to savage the architects of the 2008 financial crisis in The Big Short, wasting an A-list ensemble cast in the process. Steve Carell, Brad Pitt, Christian Bale and Ryan Gosling play various tenuously related members of the finance industry, men who made made a killing by betting against the housing market, which at that point had superficially swelled to record highs. All of the elements are in place for a lacerating satire, but almost every aesthetic choice in the film is bad, from the U-Turn-era Oliver Stone visuals to Carell's sketch-comedy performance to the cheeky cutaways where Selena Gomez and Anthony Bourdain explain complex financial concepts. After a brutal opening half, it finally settles into a groove, and there's a queasy charge in watching a credit-drunk America walking towards that cliff's edge, but not enough to save the film.

Retain words in Hu & Liu Dictionary. . .

Director and co-screenwriter Adam McKay (Step Brothers) bungles a great opportunity to savage the architects of the 2008 financial crisis in The Big Short, wasting an A-list ensemble cast in the process. Steve Carell, Brad Pitt, Christian Bale and Ryan Gosling play various tenuously related members of the finance industry, men who made made a killing by betting against the housing market, which at that point had superficially swelled to record highs. All of the elements are in place for a lacerating satire, but almost every aesthetic choice in the film is bad, from the U-Turn-era Oliver Stone visuals to Carell's sketch-comedy performance to the cheeky cutaways where Selena Gomez and Anthony Bourdain explain complex financial concepts. After a brutal opening half, it finally settles into a groove, and there's a queasy charge in watching a credit-drunk America walking towards that cliff's edge, but not enough to save the film.

Retain words in Hu & Liu Dictionary...

great *savage*
crisis *wasting*

tenuously
killing
superficially *swelled*

bad

complex
brutal

drunk
enough

Simple math...

negative 11

positive 2

total 13

tone $= \frac{2-11}{13} = \frac{-9}{13}$

Notes

Typically assume that "every word contributes isomorphically" (Young & Saroka): each word in dictionary has one of two values and sum totals matter.

But no requirement that s_m be dichotomous or integer valued: could be continuous.

e.g. might want to differentiate 'good' from 'great' from 'best'. Hard to come up with rules!

NB Tone of the document can be presented as a continuous value, or used to put documents in categories via some cutoff rule.

e.g. all documents with tone>0 are deemed 'positive'

NB Bag-of-words assn may be especially dubious for some dictionary tasks

e.g. context matters: "was `not` good" gets $+1$!

General Inquirer (selected)

Entry	Source	Positiv	Negativ	Pstv	Affil	Ngtv	Hostile	Strong	Power
ABILITY	H4Lvd	Positiv						Strong	
ABJECT	H4		Negativ						
ABLE	H4Lvd	Positiv		Pstv				Strong	
ABNORMAL	H4Lvd		Negativ			Ngtv			
ABOARD	H4Lvd								
ABOLISH	H4Lvd		Negativ			Ngtv	Hostile	Strong	Power
ABOLITION	Lvd								
ABOMINABLE	H4		Negativ					Strong	
ABRASIVE	H4		Negativ				Hostile	Strong	
ABROAD	H4Lvd								
ABRUPT	H4Lvd		Negativ			Ngtv			
ABSCOND	H4		Negativ				Hostile		
ABSENCE	H4Lvd		Negativ						
ABSENT#1	H4Lvd		Negativ						
ABSENT#2	H4Lvd								
ABSENT-MINDED	H4		Negativ						
ABSENTEE	H4		Negativ				Hostile		
ABSOLUTE#1	H4Lvd							Strong	
ABSOLUTE#2	H4Lvd							Strong	

provides dictionaries and software, which performs some stemming and disambiguation in terms of context

e.g. `ADULT` has two meanings: one is a ‘virtue’, one is a ‘role’

Dictionaries II: Linguistic Inquiry and Word Count (LIWC)

Pennebaker et al, `http://liwc.wpengine.com/`

LIWC2007 dictionary contains 2290 words and word stems (see also LIWC2015)

80 categories, organized hierarchically into 4 larger groups.

e.g. all anger words (e.g. `hate`) $\subset$ negative emotion $\subset$ affective processes $\subset$ psychological processes

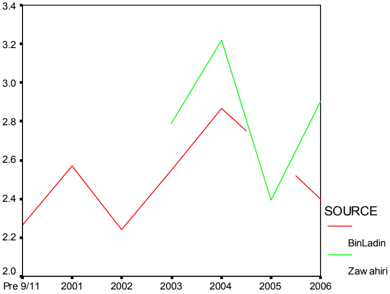
NB words can be in multiple categories, and each subdictionary score is incremented as such words appear.

Based on somewhat involved human coding/judgement and proprietary.

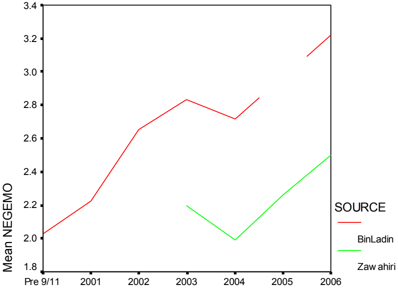
Pennebaker & Chung, 2007: Computerized Analysis of Al-Qaeda Transcripts

"The LIWC analyses suggest that Bin Ladin has been increasing in his cognitively complexity and emotionality since 9/11, as reflected by his increased use of exclusive, positive emotion, and negative emotion word use. "

C. Positive emotion (happy, love)



D. Negative emotion (hate, sad)



Dictionaries IV: Hu & Liu

2004 Hu and Liu ("Mining and Summarizing Customer Reviews") provide 6800 words which are positive and negative derived from `amazon.com` and others.

1,036 of 1,144 people found the following review helpful

★★★★★ **With Great Powers Comes Great Responsibility**

By Tommy H. on July 17, 2009

I admit it, I'm a ladies' man. And when you put this shirt on a ladies' man, it's like giving an AK-47 to a ninja. Sure it looks cool and probably would make for a good movie, but you know somebody is probably going to get hurt in the end (no pun intended). That's what almost happened to me, this is my story...

Generating New Words

Three ways to create dictionaries (non-exhaustive):

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods
- Manual generation

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods
- Manual generation
    - Careful thought (prayer? epiphanies? divine intervention?) about useful words

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
  a) Undergraduates: Pizza $\rightarrow$ Research Output

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
  a) Undergraduates: Pizza $\rightarrow$ Research Output
  b) Mechanical turkers

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
  a) Undergraduates: Pizza $\rightarrow$ Research Output
  b) Mechanical turkers
     - Example: $\{$ Happy, Unhappy $\}$

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
  a) Undergraduates: Pizza → Research Output
  b) Mechanical turkers
     - Example: { Happy, Unhappy }
     - Ask turkers: how happy is

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
  a) Undergraduates: Pizza $\rightarrow$ Research Output
  b) Mechanical turkers
     - Example: { Happy, Unhappy }
     - Ask turkers: how happy is `elevator`, `car`, `pretty`, `young`

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
  a) Undergraduates: Pizza $\rightarrow$ Research Output
  b) Mechanical turkers
     - Example: { Happy, Unhappy }
     - Ask turkers: how happy is
       `elevator`, `car`, `pretty`, `young`
       Output as dictionary

How to build a dictionary

- The ideal content analysis dictionary associates all and only the relevant words to each category in a perfectly valid scheme
- Three key issues:
  - Validity: Is the dictionary's category scheme valid?
  - Sensitivity: Does this dictionary identify all my content?
  - Specificity: Does it identify only my content?

How to build a dictionary

1. Identify “extreme texts” with “known” positions. Examples:
   - Opposition leader and Prime Minister in a no-confidence debate
   - Opposition leader and Finance Minister in a budget debate
   - Five-star review of a product (excellent) and a one-star review (terrible)
2. Search for differentially occuring words using word frequencies
3. Examine these words in context to check their sensitivity and specificity
4. Examine inflected forms to see whether stemming or wildcarding is required
5. Use these words (or their lemmas) for categories

Detecting "keywords"

- Detects words that discriminate between partitions of a corpus
- For instance, we could partition the Irish budget speech corpus into "government" and "opposition" speeches, and look for words that occur in one partition with higher relative frequency in opposition than in government speeches
- This is done by constructing a 2×2 table for each word, and testing association between that word and the partition categories

Discrimination

So Once researcher has *extreme* examples of text, various methods to identify the words that discriminate between them...

$\rightarrow$ these words then become scored as part of the dictionary/thesaurus. Can use `WordNet` to find synonyms.

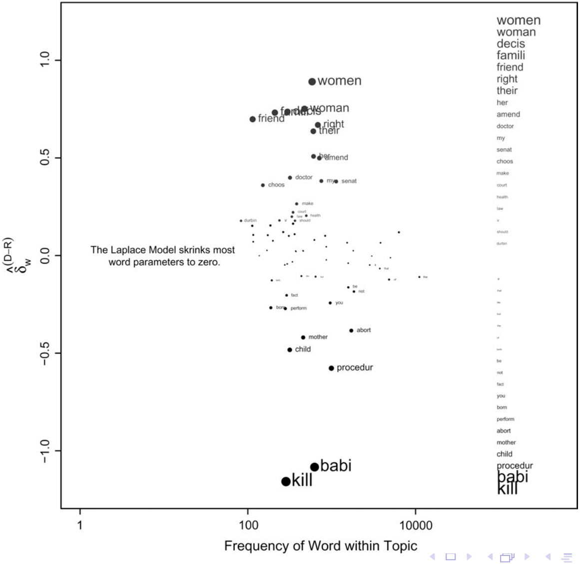
2013 Taddy provides *Multinomial Inverse Regression* to dimension reduce text, and make outcomes a product of that (reduced) set of Xs

$\rightarrow$ can be used to produce key predictors/keywords that discriminate in terms of categories.

2009 Monroe, Colaresi & Quinn consider ways to capture partisan differences in speech, and suggest Bayesian shrinkage estimator approach.

$\rightarrow$ previous approaches tend to overfit to obscure words or groups that don't have much validity in context.

Most Democratic and Republican Words on Abortion (106th, Laplace prior)



Events, dear boy...

Scholars of International Relations need access to events

Real time media reports are obvious source...


ASIA PACIFIC

Leaders of South Korea and Japan Meet in Effort to Mend Ties

Read in Chinese

By CHOE SANG-HUN NOV. 1, 2015

Yet need to be coded automatically to be helpful.

Premise and Resources

1994 Philip Schrodt develops Kansas Event Data System

2000 `TABARI` —Textual Analysis by Augmeted Replacement Instructions—open source.

also many related products, including CAMEO dealing specifically with mediation

while Virtual Research Associates Reader VRA is proprietary version.

idea first sentence of Reuters news feed (‘lead’) contains...

source of event, subject of sentence

target of event, object of sentence (direct or indirect)

type of event, transitive verb of sentence

Use and Example (Lowe & King, 2003)

> Russian artillery^{S} south of the Chechen capital Grozny blasted^{223} Chechen positions^{T} overnight before falling silent at dawn, witnesses said on Tuesday

S is the source

T is the target

223 is the code of the event between them

Hierarchical Coding Scheme (CAMEO)/Dictionary

12: REJECT

120: Reject, not specified below

121: Reject material cooperation

- 1211: Reject economic cooperation
- 1212: Reject military cooperation

122: Reject request or demand for material aid, not specified below

- 1221: Reject request for economic aid
- 1222: Reject request for military aid
- 1223: Reject request for humanitarian aid
- 1224: Reject request for military protection or peacekeeping

CAMEO	**1222**
Name	**Reject request for military aid**
Description	Refuse to extend military assistance.
Example	The Turkish government has refused to commit to any direct assistance to the US-led war against Iraq, citing domestic opposition.

Actors (CAMEO)/Dictionary

UGAREBLRA	Lord's Resistance Army
UIG	Uighur (Chinese ethnic minority)
UIS	Unidentified state actors
UKR	Ukraine
URY	Uruguay
USA	United States
USR	Union of Soviet Socialist Republics (USSR)
UZB	Uzbekistan
VAT	Holy See (Vatican City)
VCT	Saint Vincent and the Grenadines
VEN	Venezuela
VGB	British Virgin Islands

Delving More Deeply

- Begins with basic parsing: POS, stemming, stop words etc.

- Much effort to disambiguate:

  Use of pronouns causes problems.

  e.g. President is referred to as '`he`' in subsequent sentences

  Synonyms (and metonyms!) also require dictionaries (`WordNet`).

  e.g. '`US`', '`American`' ('`US`', '`Washington`')

  Care over verb/noun problems.

  e.g. '`attack`' as noun and verb

- Excellent performance relative to human coders (Lowe & King, 2003): both in terms of reliability and validity.

Summing up

Applying the model:

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}, \; (i = 1, \ldots, N)$

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}, \ (i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}, \ (i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}, \ (i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
  - $\theta_k \in \{0, 1\}$
  - $\theta_k \in \{-1, 0, 1\}$
  - $\theta_k \in \{-2, -1, 0, 1, 2\}$
  - $\theta_k \in \Re$

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}, \ (i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

For each document i calculate score for document

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

For each document i calculate score for document

$$Y_i = \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k}$$

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}, \ (i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

For each document i calculate score for document

$$
\begin{aligned}
Y_i &= \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k} \\
Y_i &= \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \mathbf{1}}
\end{aligned}
$$

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}, \; (i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
  - $\theta_k \in \{0, 1\}$
  - $\theta_k \in \{-1, 0, 1\}$
  - $\theta_k \in \{-2, -1, 0, 1, 2\}$
  - $\theta_k \in \Re$

For each document i calculate score for document

$$
\begin{aligned}
Y_i &= \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k} \\
Y_i &= \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \mathbf{1}}
\end{aligned}
$$

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}, \ (i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

For each document i calculate score for document

$$
\begin{aligned}
Y_i &= \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k} \\
Y_i &= \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \mathbf{1}}
\end{aligned}
$$

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

$Y_i > 0 \Rightarrow$ Positive Category

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}, \ (i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
  - $\theta_k \in \{0, 1\}$
  - $\theta_k \in \{-1, 0, 1\}$
  - $\theta_k \in \{-2, -1, 0, 1, 2\}$
  - $\theta_k \in \Re$

For each document i calculate score for document

$$
\begin{aligned}
Y_i &= \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k} \\
Y_i &= \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \mathbf{1}}
\end{aligned}
$$

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

$Y_i > 0 \Rightarrow$ Positive Category

$Y_i < 0 \Rightarrow$ Negative Category

Summing up

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK}, \ (i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
  - $\theta_k \in \{0, 1\}$
  - $\theta_k \in \{-1, 0, 1\}$
  - $\theta_k \in \{-2, -1, 0, 1, 2\}$
  - $\theta_k \in \Re$

For each document i calculate score for document

$$\begin{aligned} Y_i &= \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k} \\ Y_i &= \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \mathbf{1}} \end{aligned}$$

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

$Y_i > 0 \Rightarrow$ Positive Category

$Y_i < 0 \Rightarrow$ Negative Category

$Y_i \approx 0$ Ambiguous

Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

- No optimization step $\rightsquigarrow$ same word weights regardless of texts

Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

- No optimization step $\rightsquigarrow$ same word weights regardless of texts
- Optimization$\rightsquigarrow$ incorporate information specific to context

Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

- No optimization step $\rightsquigarrow$ same word weights regardless of texts
- Optimization$\rightsquigarrow$ incorporate information specific to context
- Without optimization$\rightsquigarrow$ unclear about dictionaries performance

Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

- No optimization step $\rightsquigarrow$ same word weights regardless of texts
- Optimization$\rightsquigarrow$ incorporate information specific to context
- Without optimization$\rightsquigarrow$ unclear about dictionaries performance

Just because dictionaries provide measures labeled "positive" or "negative" it doesn't mean they are accurate measures in your text (!!!!)

Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

- No optimization step $\rightsquigarrow$ same word weights regardless of texts
- Optimization$\rightsquigarrow$ incorporate information specific to context
- Without optimization$\rightsquigarrow$ unclear about dictionaries performance

Just because dictionaries provide measures labeled "positive" or "negative" it doesn't mean they are accurate measures in your text (!!!!)

Validation

Being Careful...

In principle, it is straightforward to extend dictionary from one domain to another;

$\rightarrow$ matter of adding extra words in the various categories.

But much care is needed when a dictionary designed for one context is applied to another.

e.g. Loughran & MacDonald, 2011: common dictionaries fail badly when applied to financial texts. e.g. cost is a neutral term in reports, but negative in Harvard IV

plus virtually impossible to validate dictionaries: very expensive, at least.

btw humans not very good at producing discriminating terms for e.g. opinion mining (Pang et al, 2002)

Validation, Dictionaries from other Fields

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

Loughran and McDonald (2011): Financial Documents are Different, polysemes

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

Loughran and McDonald (2011): Financial Documents are Different, polysemes

- Negative words in Harvard, Not Negative in Accounting:

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

Loughran and McDonald (2011): Financial Documents are Different, polysemes

- Negative words in Harvard, Not Negative in Accounting: `tax,cost, capital, board, liability, foreign, cancer, crude (oil),tire`

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

Loughran and McDonald (2011): Financial Documents are Different, polysemes

- Negative words in Harvard, Not Negative in Accounting: `tax,cost, capital, board, liability, foreign, cancer, crude (oil),tire`
- 73% of Harvard negative words in this set(!!!!!)

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

Loughran and McDonald (2011): Financial Documents are Different, polysemes

- Negative words in Harvard, Not Negative in Accounting: `tax,cost, capital, board, liability, foreign, cancer, crude (oil),tire`
- 73% of Harvard negative words in this set(!!!!!)
- Not Negative Harvard, Negative in Accounting:

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts
Loughran and McDonald (2011): Financial Documents are Different, polysemes

- Negative words in Harvard, Not Negative in Accounting: `tax,cost, capital, board, liability, foreign, cancer, crude (oil),tire`
- 73% of Harvard negative words in this set(!!!!!)
- Not Negative Harvard, Negative in Accounting: `felony, litigation, restated, misstatement, and unanticipated`

Validation

Classification Validity:

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?
  - Is the dictionary your using appropriate?

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
    - Is the classification scheme well defined for your texts?
    - Can humans accomplish the coding task?
    - Is the dictionary your using appropriate?

Replicate classification exercise

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?
  - Is the dictionary your using appropriate?

Replicate classification exercise

- How well does our method perform on held out documents?

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?
  - Is the dictionary your using appropriate?

Replicate classification exercise

- How well does our method perform on held out documents?
- Why held out?

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?
  - Is the dictionary your using appropriate?

Replicate classification exercise

- How well does our method perform on held out documents?
- Why held out? Over fitting

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
    - Is the classification scheme well defined for your texts?
    - Can humans accomplish the coding task?
    - Is the dictionary your using appropriate?

Replicate classification exercise

- How well does our method perform on held out documents?
- Why held out? Over fitting
- Using off-the-shelf dictionary: all labeled documents to test

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?
  - Is the dictionary your using appropriate?

Replicate classification exercise

- How well does our method perform on held out documents?
- Why held out? Over fitting
- Using off-the-shelf dictionary: all labeled documents to test
- Supervised learning classification: (Cross)validation

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?
  - Ambiguity in language

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?
    - Ambiguity in language
    - Limited working memory

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?
  - Ambiguity in language
  - Limited working memory
  - Ambiguity in classification rules

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?
  - Ambiguity in language
  - Limited working memory
  - Ambiguity in classification rules
- A procedure for training coders:

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?
  - Ambiguity in language
  - Limited working memory
  - Ambiguity in classification rules
- A procedure for training coders:
  1) Coding rules
  2) Apply to new texts
  3) Assess coder agreement (we'll discuss more in a few weeks)
  4) Using information and discussion, revise coding rules

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

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

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

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

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

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

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

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

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

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

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

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

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

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

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

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

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

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

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

Under reported for dictionary classification

What about continuous measures?

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement
- Precisely the point $\rightsquigarrow$ merely creating a gold standard is hard, let alone computer classification

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement
- Precisely the point $\rightsquigarrow$ merely creating a gold standard is hard, let alone computer classification

Lower level classification

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement
- Precisely the point $\rightsquigarrow$ merely creating a gold standard is hard, let alone computer classification

Lower level classification $\rightsquigarrow$ label phrases and then aggregate

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement
- Precisely the point$\rightsquigarrow$ merely creating a gold standard is hard, let alone computer classification

Lower level classification$\rightsquigarrow$ label phrases and then aggregate
Modifiable areal unit problem in texts$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement
- Precisely the point $\rightsquigarrow$ merely creating a gold standard is hard, let alone computer classification

Lower level classification $\rightsquigarrow$ label phrases and then aggregate
Modifiable areal unit problem in texts $\rightsquigarrow$ aggregating destroys information, conclusion may depend on level of aggregation

Supervised Learning

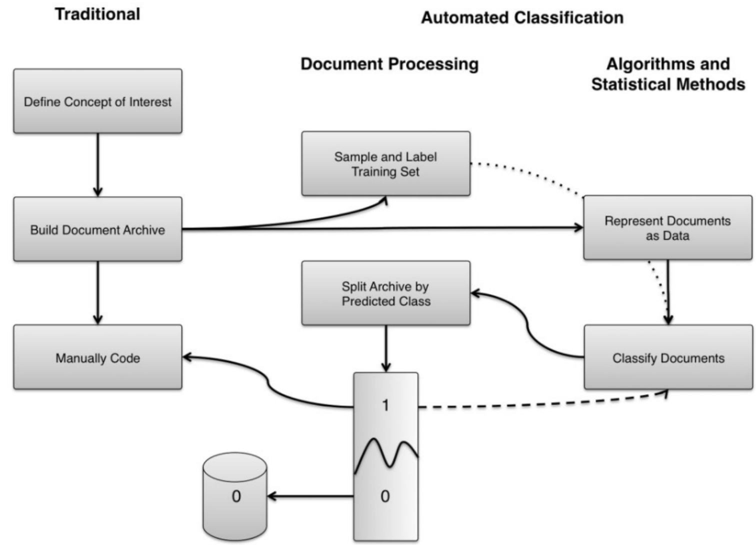


Fig. 1 The data collection process.

Supervised Learning

Clustering and Topic Models:

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

Supervised Learning

Clustering and Topic Models:

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

Supervised Methods:

Supervised Learning

Clustering and Topic Models:

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

Supervised Methods:

- Models for categorizing texts

Supervised Learning

Clustering and Topic Models:

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

Supervised Methods:

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

Supervised Learning

Clustering and Topic Models:

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

Supervised Methods:

- 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

Clustering and Topic Models:

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

Supervised Methods:

- 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

Recap - Components to Supervised Learning Method

Recap - Components to Supervised Learning Method

1) Set of categories

Recap - Components to Supervised Learning Method

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

Recap - 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

Recap - 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

Recap - 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

Recap - 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.

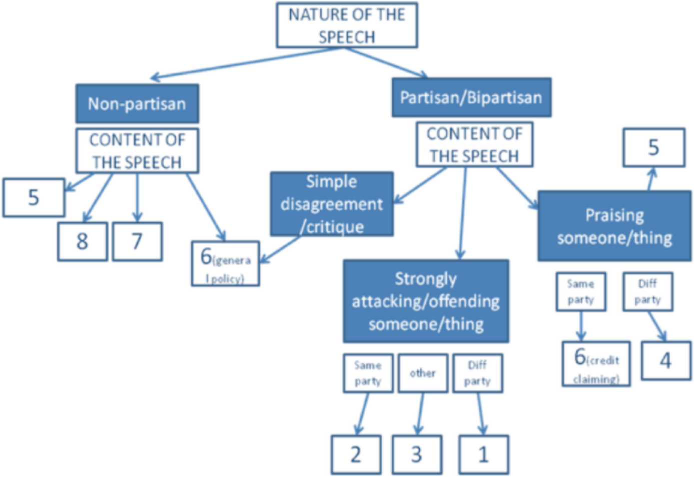
	Coder C								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	Tota
Coder B									
1	**18**	0	1	0	0	0	0	0	
3	1	0	**1**	0	0	0	0	0	
4	0	0	1	**7**	0	1	0	0	
5	0	0	0	2	**18**	3	0	0	
6	13	1	7	4	1	**26**	0	0	
7	3	0	0	0	0	8	**63**	2	
8	0	0	0	0	0	4	1	**15**	
Total	35	1	10	13	19	42	64	17	

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). ($\Pr(\text{Class } 1) = 0.75$, $\Pr(\text{Class } 2) = 0.25$)
- 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). ($\Pr(\text{Class 1}) = 0.75$, $\Pr(\text{Class 2}) = 0.25$)
- 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). ($\Pr(\text{Class 1}) = 0.75$, $\Pr(\text{Class 2}) = 0.25$)
- 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:

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

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). ($\Pr(\text{Class 1}) = 0.75$, $\Pr(\text{Class 2}) = 0.25$)
- Inter Coder reliability: 0.625

What to do?

Suggestion: Subtract off amount expected by chance:

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

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:

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

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). ($\Pr(\text{Class 1}) = 0.75$, $\Pr(\text{Class 2}) = 0.25$)
- Inter Coder reliability: 0.625

What to do?

Suggestion: Subtract off amount expected by chance:

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

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

Define coder reliability as:

Krippendorf's Alpha

Define coder reliability as:

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

Krippendorf's Alpha

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

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

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

Thinking through expected differences:

Krippendorf's Alpha

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

Thinking through expected differences:

- 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

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

Thinking through expected differences:

- 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

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

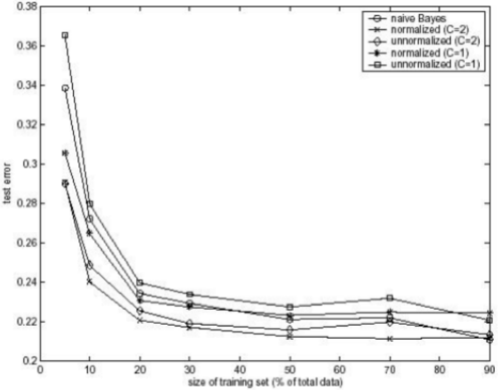


Figure 2: Test error vs training size on the newsgroups alt.atheism and talk.religion.misc

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

- Use the hand labels to train a statistical model.
- Naive Bayes
  - Shockingly simple application of Bayes' rule
  - Shockingly useful $\rightsquigarrow$ often default classifier

Naive Bayes and General Problem Setup

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

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k $(k = 1, \ldots, K)$
$\{C_1, C_2, \ldots, C_K\}$

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k $(k = 1, \ldots, K)$
$\{C_1, C_2, \ldots, C_K\}$
Subset of labeled documents $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_{N_{\text{train}}})$ where
$Y_i \in \{C_1, C_2, \ldots, C_K\}$.

Naive Bayes and General Problem Setup

Suppose we have document i, ($i = 1, \ldots, N$) with J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k ($k = 1, \ldots, K$)
$\{C_1, C_2, \ldots, C_K\}$
Subset of labeled documents $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_{N_{\text{train}}})$ where
$Y_i \in \{C_1, C_2, \ldots, C_K\}$.
Goal: classify every document into one category.

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k $(k = 1, \ldots, K)$
$\{C_1, C_2, \ldots, C_K\}$
Subset of labeled documents $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_{N_{\text{train}}})$ where
$Y_i \in \{C_1, C_2, \ldots, C_K\}$.
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 J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k $(k = 1, \ldots, K)$
$\{C_1, C_2, \ldots, C_K\}$
Subset of labeled documents $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_{N_{\text{train}}})$ where
$Y_i \in \{C_1, C_2, \ldots, C_K\}$.
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 J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k $(k = 1, \ldots, K)$
$\{C_1, C_2, \ldots, C_K\}$
Subset of labeled documents $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_{N_{\text{train}}})$ where
$Y_i \in \{C_1, C_2, \ldots, C_K\}$.
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

Reminder: Bayes' Theorem

Recall that:

$$\Pr(A|B) = \frac{\Pr(A,B)}{\Pr(B)}$$

- the probability that A occurs given that B occurred = the probability of both A and B occurring, divided by the probability that B occurs.

e.g. you know a die shows an odd number, what is the probability that this odd number is 3? $\Pr(3|\text{odd}) = \frac{\frac{1}{6}}{\frac{1}{2}} = \frac{1}{3}$.

- of course, it is also true that $\Pr(B|A) = \frac{\Pr(B,A)}{\Pr(A)}$.
- but then, since $\Pr(A,B) = \Pr(B,A)$, we must have $\Pr(A|B)\Pr(B) = \Pr(B|A)\Pr(A)$, and thus... Bayes' law

$$\boxed{\Pr(A|B) = \frac{\Pr(A)\Pr(B|A)}{\Pr(B)}.}$$

And. . .

- interest is in $\Pr(A|B) = \frac{\Pr(A)\Pr(B|A)}{\Pr(B)}$.
- Notice that $\Pr(B)$ itself does not tell us whether a particular value of A is more or less likely to be observed, so drop it and rewrite:

$$\Pr(A|B) \propto \Pr(A)\Pr(B|A)$$

Here, $\Pr(A)$ is our prior for A, while $\Pr(B|A)$ will be the likelihood for the data we saw.

So...

Given $c =$ class and $d =$ document, $p(c|d) = \frac{p(d|c)p(c)}{p(d)}$

- $p(c|d) =$ probability of instance d being in class c, This is what we are trying to compute
- $p(d|c) =$ probability of generating instance d given class c. We can imagine that being in class c, causes you to have feature d with some probability
- $p(c) =$ probability of occurrence of class c. This is just how frequent the class c, is in our data
- $p(d) =$ probability of instance d occurring. This can actually be ignored, since it is the same for all classes

Reformulate the problem at the word level...

Consider J word types distributed across I documents, each assigned one of K classes.

At the word level, Bayes Theorem tells us that:

$$P(c_k|w_j) = \frac{P(w_j|c_k)P(c_k)}{P(w_j)}$$

For two classes, this can be expressed as

$$= \frac{P(w_j|c_k)P(c_k)}{P(w_j|c_k)P(c_k) + P(w_j|c_{\neg k})P(c_{\neg k})}$$

Class-conditional word likelihoods

$$P(c_k|w_j) = \frac{P(w_j|c_k)P(c_k)}{P(w_j|c_k)P(c_k) + P(w_j|c_{\neg k})P(c_{\neg k})}$$

- The word likelihood within class
- The maximum likelihood estimate is simply the proportion of times that word j occurs in class k, but it is more common to use Laplace smoothing by adding 1 to each oberved count within class

Word probabilities

$$P(c_k|w_j) = \frac{P(w_j|c_k)P(c_k)}{P(w_j)}$$

- This represents the word probability from the training corpus
- Usually uninteresting, since it is constant for the training data, but needed to compute posteriors on a probability scale

Class prior probabilities

$$P(c_k|w_j) = \frac{P(w_j|c_k)P(c_k)}{P(w_j|c_k)P(c_k) + P(w_j|c_{\neg k})P(c_{\neg k})}$$

- This represents the class prior probability
- Machine learning typically takes this as the document frequency in the training set
- This approach is flawed for scaling, however, since we are scaling the latent class-ness of an unknown document, not predicting class – uniform priors are more appropriate

Class posterior probabilities

$$P(c_k|w_j) = \frac{P(w_j|c_k)P(c_k)}{P(w_j|c_k)P(c_k) + P(w_j|c_{\neg k})P(c_{\neg k})}$$

- This represents the posterior probability of membership in class k for word j

Moving to the document level

- The "Naive" Bayes model of a joint document-level class posterior assumes conditional independence, to multiply the word likelihoods from a "test" document, to produce:

$$P(c|d) = P(c) \prod_j \frac{P(w_j|c)}{P(w_j)}$$

- This is why we call it "naive": because it (wrongly) assumes:
  - *conditional independence* of word counts
  - *positional independence* of word counts

Example

	email	words	classification
training	1	money inherit prince	spam
	2	prince inherit amount	spam
	3	inherit plan money	ham
	4	cost amount amazon	ham
	5	prince william news	ham
test	6	prince prince money	?

$$\Pr(\text{prince}|\text{ham}) = \frac{1}{9}$$
$$\Pr(\text{prince}|\text{ham}) = \frac{1}{9}$$
$$\Pr(\text{money}|\text{ham}) = \frac{1}{9}$$
$$\Pr(\text{ham}|d) \propto \frac{3}{5}\frac{1}{9}\frac{1}{9}\frac{1}{9} = 0.00082$$

$$\Pr(\text{prince}|\text{spam}) = \frac{2}{6}$$
$$\Pr(\text{prince}|\text{spam}) = \frac{2}{6}$$
$$\Pr(\text{money}|\text{spam}) = \frac{1}{6}$$
$$\Pr(\text{spam}|d) \propto \frac{2}{5}\frac{2}{6}\frac{2}{6}\frac{1}{6} = 0.0074$$

$$\rightarrow \boxed{c_{map} = \text{spam}}$$

Assume that we have two classes

c_1 = male, and c_2 = female.

We have a person whose sex we do not know, say "*drew*" or *d*.

Classifying *drew* as male or female is equivalent to asking is it more probable that *drew* is male or female, I.e which is greater $p(\text{male} \mid drew)$ or $p(\text{female} \mid drew)$

(Note: "Drew can be a male or female name")

Drew Barrymore

Drew Carey

What is the probability of being called "*drew*" given that you are a male?

What is the probability of being a male?

$$p(\text{male} \mid drew) = \frac{p(drew \mid \text{male})\, p(\text{male})}{p(drew)}$$

What is the probability of being named "*drew*"?

(actually irrelevant, since it is that same for all classes)

Officer Drew

This is Officer Drew (who arrested me in 1997). Is Officer Drew a Male or Female?

Luckily, we have a small database with names and sex.

We can use it to apply Bayes rule…

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

$$p(c_j \mid d) = \frac{p(d \mid c_j)\, p(c_j)}{p(d)}$$

Officer Drew

$$p(c_j \mid d) = \frac{p(d \mid c_j)\, p(c_j)}{p(d)}$$

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

$$p(\text{male} \mid drew) = \frac{1/3 * 3/8}{3/8} = \frac{0.125}{3/8}$$

$$p(\text{female} \mid drew) = \frac{2/5 * 5/8}{3/8} = \frac{0.250}{3/8}$$

Officer Drew is more likely to be a Female.

Officer Drew

Officer Drew IS a female!

$$p(\text{male} \mid drew) = \frac{1/3 * 3/8}{3/8} = \frac{0.125}{3/8}$$

$$p(\text{female} \mid drew) = \frac{2/5 * 5/8}{3/8} = \frac{0.250}{3/8}$$

What about multiple features?

Name	Over 170CM	Eye	Hair length	Sex
Drew	No	Blue	Short	Male
Claudia	Yes	Brown	Long	Female
Drew	No	Blue	Long	Female
Drew	No	Blue	Long	Female
Alberto	Yes	Brown	Short	Male
Karin	No	Blue	Long	Female
Nina	Yes	Brown	Short	Female
Sergio	Yes	Blue	Long	Male

Without loss of generalization, we can represent a document d as a set of features $f_1, f_2, ..., f_n$:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(f_1, f_2, ..., f_n|c)}^{\text{likelihood}} \ \overbrace{P(c)}^{\text{prior}}$$

Two core assumptions

- Bag of Words assumption: we assume word position doesn?t matter, and that the word "love" has the same effect on classification whether it occurs as the 1st, 20th, or last word in the document. Thus we assume that the features $f_1, f_2, \ldots, f_n$ only encode word identity and not position. The prob a term occurs in a particular place is constant for entire document, which means we only need one probability distribution of terms that is valid for every position.
- Conditional Independence assumption: that the probabilities $P(f_i|c)$ are indedpendent give the class, and hence can be "naively" multiplied as follows $P(f_1, f_2, \ldots, f_n|c) = P(f_1|c) * P(f_2|c) * \ldots * P(f_n|c)$. That is, once we condition on a given category, the probability that a particular word occurs is independent of any other feature occurring.

- To simplify the task, **naïve Bayesian classifiers** assume attributes have independent distributions, and thereby estimate

$$p(d|c_j) = p(d_1|c_j) * p(d_2|c_j) * \ldots.* p(d_n|c_j)$$

The probability of class c_j generating instance d, equals….

The probability of class c_j generating the observed value for feature 1, multiplied by..

The probability of class c_j generating the observed value for feature 2, multiplied by..

- To simplify the task, **naïve Bayesian classifiers** assume attributes have independent distributions, and thereby estimate

$$p(d|c_j) = p(d_1|c_j) * p(d_2|c_j) * \ldots.* p(d_n|c_j)$$

$$p(\text{officer drew}|c_j) = p(\text{over\_170}_{\text{cm}} = \text{yes}|c_j) * p(\text{eye} = blue|c_j) * \ldots.$$

Officer Drew is blue-eyed, over 170_{cm} tall, and has long hair

$$p(\text{officer drew}|\ \text{Female}) = 2/5 \quad * \quad 3/5 \quad * \quad \ldots.$$
$$p(\text{officer drew}|\ \text{Male}) = 2/3 \quad * \quad 2/3 \quad * \quad \ldots.$$

Training Naive Bayes

function TRAIN NAIVE BAYES(D, C) **returns** log $P(c)$ and log $P(w|c)$

for each class $c \in C$ # Calculate $P(c)$ terms
  N_{doc} = number of documents in D
  N_c = number of documents from D in class c
  $logprior[c] \leftarrow \log \frac{N_c}{N_{doc}}$
  $V \leftarrow$ vocabulary of D
  $bigdoc[c] \leftarrow$ **append**(d) **for** d $\in$ D **with** class c
  for each word w in V # Calculate $P(w|c)$ terms
    $count(w,c) \leftarrow$ # of occurrences of w in $bigdoc[c]$
    $loglikelihood[w,c] \leftarrow \log \frac{count(w,c) + 1}{\sum_{w' \text{ in } V} (count(w',c) + 1)}$
return *logprior*, *loglikelihood*, *V*

function TEST NAIVE BAYES(*testdoc*, *logprior*, *loglikelihood*, C, V) **returns** best c

for each class $c \in C$
  $sum[c] \leftarrow logprior[c]$
  for each position i in *testdoc*
    $word \leftarrow testdoc[i]$
    if $word \in V$
      $sum[c] \leftarrow sum[c] + loglikelihood[word,c]$
return $\text{argmax}_c\ sum[c]$

Naive Bayes and General Problem Setup

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

(0.1)

Naive Bayes and General Problem Setup

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

$$C_{\text{Max}} = \arg\max_k p(C_k|\boldsymbol{x}_i) \qquad (0.1)$$

Naive Bayes and General Problem Setup

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

$$C_{\text{Max}} = \arg\max_k p(C_k|\boldsymbol{x}_i) \tag{0.1}$$

We're going to use Bayes' rule to estimate $p(C_k|\boldsymbol{x}_i)$.

Naive Bayes and General Problem Setup

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

$$C_{\text{Max}} = \arg\max_k p(C_k|\boldsymbol{x}_i)$$

We're going to use Bayes' rule to estimate $p(C_k|\boldsymbol{x}_i)$.

$$p(C_k|\boldsymbol{x}_i) = \frac{p(C_k, \boldsymbol{x}_i)}{p(\boldsymbol{x}_i)} \tag{0.1}$$

Naive Bayes and General Problem Setup

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

$$C_{\text{Max}} = \arg\max_k p(C_k|\boldsymbol{x}_i)$$

We're going to use Bayes' rule to estimate $p(C_k|\boldsymbol{x}_i)$.

$$\begin{aligned} p(C_k|\boldsymbol{x}_i) &= \frac{p(C_k, \boldsymbol{x}_i)}{p(\boldsymbol{x}_i)} \\ &= \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \end{aligned} \tag{0.1}$$

Naive Bayes and General Problem Setup

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

$$C_{\text{Max}} \quad = \quad \arg\max_k p(C_k|\boldsymbol{x}_i)$$

We're going to use Bayes' rule to estimate $p(C_k|\boldsymbol{x}_i)$.

$$\begin{aligned} p(C_k|\boldsymbol{x}_i) &= \frac{p(C_k, \boldsymbol{x}_i)}{p(\boldsymbol{x}_i)} \\ &= \frac{\overbrace{p(C_k)}^{\text{Proportion in } C_k} \underbrace{p(\boldsymbol{x}_i|C_k)}_{\text{Language model}}}{p(\boldsymbol{x}_i)} \end{aligned}$$

Naive Bayes and Optimization

Naive Bayes and Optimization

$$C_{\text{Max}} \quad = \quad \arg\max_k \; p(C_k | \boldsymbol{x}_i)$$

Naive Bayes and Optimization

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_k \; p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \arg\max_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)}
\end{aligned}
$$

Naive Bayes and Optimization

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_k \; p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \arg\max_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\
C_{\text{Max}} &= \arg\max_k \; p(C_k)p(\boldsymbol{x}_i|C_k)
\end{aligned}
$$

Naive Bayes and Optimization

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_k \; p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \arg\max_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\
C_{\text{Max}} &= \arg\max_k \; p(C_k)p(\boldsymbol{x}_i|C_k)
\end{aligned}
$$

Two probabilities to estimate:

Naive Bayes and Optimization

$$\begin{aligned} C_{\text{Max}} &= \arg\max_k \; p(C_k|\boldsymbol{x}_i) \\ C_{\text{Max}} &= \arg\max_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\ C_{\text{Max}} &= \arg\max_k \; p(C_k)p(\boldsymbol{x}_i|C_k) \end{aligned}$$

Two probabilities to estimate:

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

Naive Bayes and Optimization

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_k \; p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \arg\max_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\
C_{\text{Max}} &= \arg\max_k \; p(C_k)p(\boldsymbol{x}_i|C_k)
\end{aligned}
$$

Two probabilities to estimate:

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

$p(\boldsymbol{x}_i|C_k)$ complicated without assumptions

Naive Bayes and Optimization

$$\begin{aligned} C_{\text{Max}} &= \arg\max_k \; p(C_k|\boldsymbol{x}_i) \\ C_{\text{Max}} &= \arg\max_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\ C_{\text{Max}} &= \arg\max_k \; p(C_k)p(\boldsymbol{x}_i|C_k) \end{aligned}$$

Two probabilities to estimate:

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

$p(\boldsymbol{x}_i|C_k)$ complicated without assumptions

- Imagine each x_{ij} just binary indicator. Then 2^J possible $\boldsymbol{x}_i$ documents

Naive Bayes and Optimization

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_k \ p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \arg\max_k \ \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\
C_{\text{Max}} &= \arg\max_k \ p(C_k)p(\boldsymbol{x}_i|C_k)
\end{aligned}
$$

Two probabilities to estimate:

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

$p(\boldsymbol{x}_i|C_k)$ complicated without assumptions

- Imagine each x_{ij} just binary indicator. Then 2^J possible $\boldsymbol{x}_i$ documents
- Simplify: assume each feature is independent

Naive Bayes and Optimization

$$\begin{aligned} C_{\text{Max}} &= \text{arg max}_k \ p(C_k|\boldsymbol{x}_i) \\ C_{\text{Max}} &= \text{arg max}_k \ \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\ C_{\text{Max}} &= \text{arg max}_k \ p(C_k)p(\boldsymbol{x}_i|C_k) \end{aligned}$$

Two probabilities to estimate:

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

$p(\boldsymbol{x}_i|C_k)$ complicated without assumptions

- Imagine each x_{ij} just binary indicator. Then 2^J possible $\boldsymbol{x}_i$ documents
- Simplify: assume each feature is independent

$$p(\boldsymbol{x}_i|C_k) = \prod_{j=1}^{J} p(x_{ij}|C_k)$$

Naive Bayes and Optimization

Two components to estimation:

- $p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)
- $p(\boldsymbol{x}_i | C_k) = \prod_{j=1}^{J} p(x_{ij} | C_k)$

Naive Bayes and Optimization

Two components to estimation:

- $p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)
- $p(\boldsymbol{x}_i | C_k) = \prod_{j=1}^{J} p(x_{ij} | C_k)$

Maximum likelihood estimation (training set):

Naive Bayes and Optimization

Two components to estimation:

- $p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)
- $p(\boldsymbol{x}_i | C_k) = \prod_{j=1}^{J} p(x_{ij} | C_k)$

Maximum likelihood estimation (training set):

$$p(x_{im} = z | C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)}}{\text{No(C}= \text{C}_k)}$$

Naive Bayes and Optimization

Two components to estimation:

- $p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)
- $p(\boldsymbol{x}_i | C_k) = \prod_{j=1}^{J} p(x_{ij}|C_k)$

Maximum likelihood estimation (training set):

$$p(x_{im} = z|C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)}}{\text{No(C} = \text{C}_k\text{)}}$$

Problem: What if No($\text{Docs}_{ij} = \text{z and C} = \text{C}_k$) $= 0$?

Naive Bayes and Optimization

Two components to estimation:

- $p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)
- $p(\boldsymbol{x}_i | C_k) = \prod_{j=1}^{J} p(x_{ij}|C_k)$

Maximum likelihood estimation (training set):

$$p(x_{im} = z | C_k) = \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)}}{\text{No(C} = \text{C}_k\text{)}}$$

Problem: What if No($\text{Docs}_{ij} = \text{z and C} = \text{C}_k$) $= 0$?
$\prod_{j=1}^{J} p(x_{ij}|C_k) = 0$

Naive Bayes and General Problem Setup

Naive Bayes and General Problem Setup

Solution: smoothing (Bayesian estimation)

Naive Bayes and General Problem Setup

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z | C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)} + 1}{\text{No(C} = \text{C}_k) + k}$$

Naive Bayes and General Problem Setup

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z|C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)} + 1}{\text{No(C} = \text{C}_k) + k}$$

Algorithm steps:

Naive Bayes and General Problem Setup

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z | C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)} + 1}{\text{No(C}= \text{C}_k) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\boldsymbol{x}_i | C_k)$ on training data

Naive Bayes and General Problem Setup

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z | C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)} + 1}{\text{No(C} = \text{C}_k) + k}$$

Algorithm steps:

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

Naive Bayes and General Problem Setup

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z | C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k\text{)} + 1}{\text{No(C}= \text{C}_k) + k}$$

Algorithm steps:

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

$$C_i \quad = \quad \text{arg max }_k \hat{p}(C_k) \hat{p}(\boldsymbol{x}_i | C_k)$$

Naive Bayes and General Problem Setup

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z|C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)} + 1}{\text{No(C= C}_k) + k}$$

Algorithm steps:

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

$$C_i \quad = \quad \text{arg max }_k \hat{p}(C_k)\hat{p}(\boldsymbol{x}_i|C_k)$$

Simple intuition about Naive Bayes:

Naive Bayes and General Problem Setup

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z | C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)} + 1}{\text{No(C} = \text{C}_k) + k}$$

Algorithm steps:

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

$$C_i \quad = \quad \text{arg max }_k \hat{p}(C_k)\hat{p}(\boldsymbol{x}_i | C_k)$$

Simple intuition about Naive Bayes:

- Learn what documents in class j look like

Naive Bayes and General Problem Setup

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z | C_k) = \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)} + 1}{\text{No(C= C}_k) + k}$$

Algorithm steps:

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

$$C_i = \text{arg max }_k \hat{p}(C_k)\hat{p}(\boldsymbol{x}_i | C_k)$$

Simple intuition about Naive Bayes:

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

Naive Bayes and Unigram Language Models

The probability a new document has $\tau_{ik} = 1$ is then

Naive Bayes and Unigram Language Models

The probability a new document has $\tau_{ik} = 1$ is then

$$p(\tau_{ik} = 1 | \boldsymbol{x}_i, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\theta}}) \quad \propto \quad p(\tau_{ik} = 1) p(\boldsymbol{x}_i | \boldsymbol{\theta}, \tau_{ik} = 1)$$

Naive Bayes and Unigram Language Models

The probability a new document has $\tau_{ik} = 1$ is then

$$
\begin{aligned}
p(\tau_{ik} = 1 | \boldsymbol{x}_i, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\theta}}) &\propto p(\tau_{ik} = 1) p(\boldsymbol{x}_i | \boldsymbol{\theta}, \tau_{ik} = 1) \\
&\propto \widehat{\pi_k} \prod_{j=1}^{J} \left(\widehat{\theta}_{jk} \right)^{x_{ij}}
\end{aligned}
$$

Naive Bayes and Unigram Language Models

The probability a new document has $\tau_{ik} = 1$ is then

$$\begin{aligned}
p(\tau_{ik} = 1 | \boldsymbol{x}_i, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\theta}}) &\propto p(\tau_{ik} = 1) p(\boldsymbol{x}_i | \boldsymbol{\theta}, \tau_{ik} = 1) \\
&\propto \widehat{\pi_k} \prod_{j=1}^{J} \left(\widehat{\theta}_{jk}\right)^{x_{ij}} \\
&\propto \overbrace{\widehat{\pi_k}}^{p(C_k)} \underbrace{\prod_{j=1}^{J} \left(\widehat{\theta}_{jk}\right)^{x_{ij}}}_{\text{Unigram model}}
\end{aligned}$$

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 (cross-validation, tomorrow)
- 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 Dictionary week (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 Dictionary week (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 Dictionary week (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 Dictionary week (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 Dictionary week (along with some new ones)

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

$$
\begin{aligned}
\text{Accuracy} &= \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}} \\
\text{Precision}_{\text{Liberal}} &= \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Liberal}} \\
\text{Recall}_{\text{Liberal}} &= \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Conservative}} \\
F_{\text{Liberal}} &= \frac{2\text{Precision}_{\text{Liberal}}\text{Recall}_{\text{Liberal}}}{\text{Precision}_{\text{Liberal}} + \text{Recall}_{\text{Liberal}}}
\end{aligned}
$$

Comparing Training and Validation Set

Representation of Test Statistics from Dictionary week (along with some new ones)

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

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

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

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

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

ROC Curve

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:

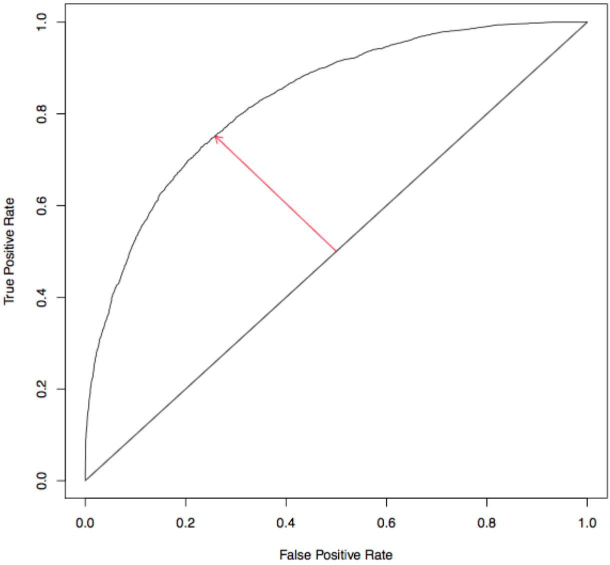
- Everything liberal: $\text{Recall}_{\text{Liberal}}$ =1 ; $\text{Recall}_{\text{Conservative}} = 0$
- Everything conservative: $\text{Recall}_{\text{Liberal}}$ =0 ; $\text{Recall}_{\text{Conservative}} = 1$

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 Grimmer work on Senate press releases

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

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

Example: Jihadi Clerics

Indonesian cleric's support for ISIS increases the security threat

July 20, 2014 10.14pm EDT

Noor Huda Ismail
PhD Candidate in Politics and International Relations , Monash University

Nielsen (2012) investigates why certain scholars of Islam become Jihadi: i.e. why they encourage armed struggle (especially against the west)

Requires that he first classifies scholars as Jihadi and ¬ Jihadi: has 27,142 texts from 101 clerics, and difficult to do by hand.

Jihadi Clerics

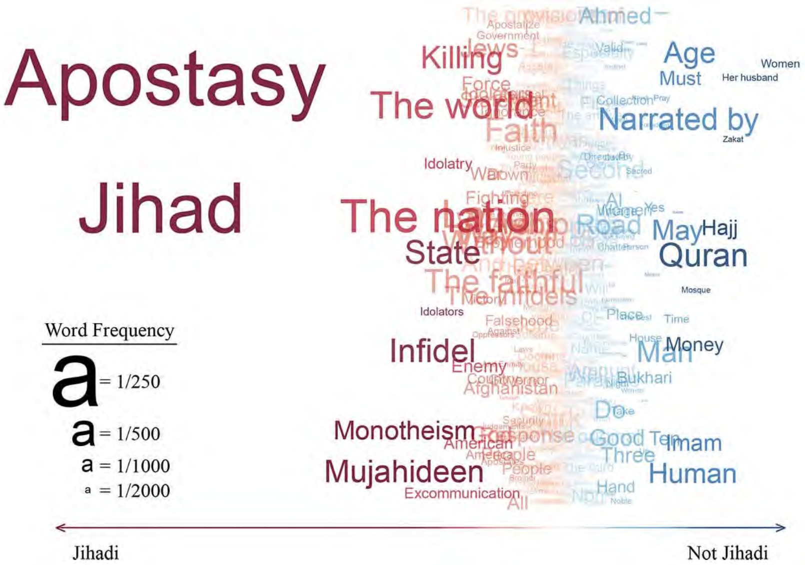
Training set: self-identified Jihadi texts (765), and sample from Islamic website as $\neg$ Jihadi (1951)

Preprocess: drops terms occurring in less than 10%, or more than 40% of documents, and uses 'light' stemmer for Arabic

Can assign a *Jihad Score* to each document: basically the logged likelihood ratio, $\sum_i \log \frac{\Pr(t_k|\text{Jihad})}{\Pr(t_k|\neg \text{ Jihad})}$ (note: doesn't know what 'real world' priors are, so drops them here)

Then for each cleric, concatenate all works into one and give this 'document'/cleric a score.

Discriminating Words



Validation: *Exoneration*

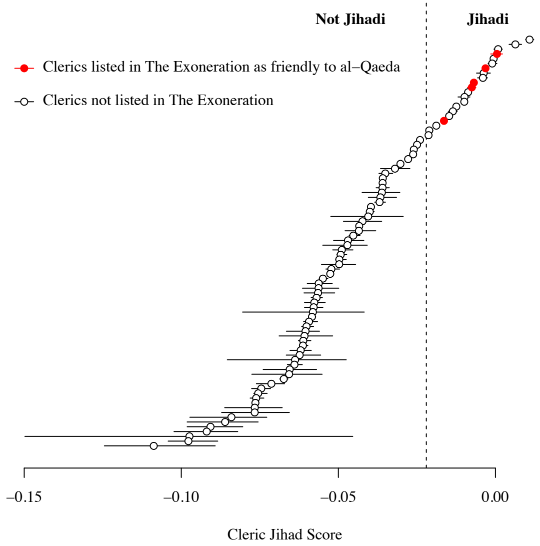


Figure 4.9: *Jihad Scores Predict Inclusion in The Exoneration*

A word on Support Vector Machines...

back to the vector space model of text...

- Suppose you have two classes: vacations and sports
- Suppose you have four documents

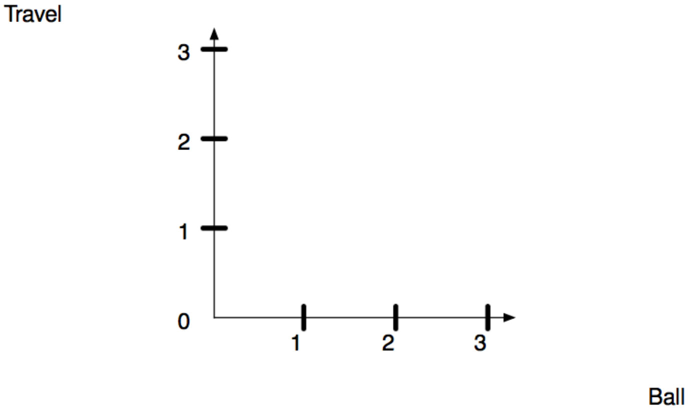
Sports
Doc_1: {ball, ball, ball, travel}
Doc_2: {ball, ball}

Vacations
Doc_3: {travel, ball, travel}
Doc_4: {travel}

- Suppose you have four documents

A word on Support Vector Machines...

Put the documents in vector space



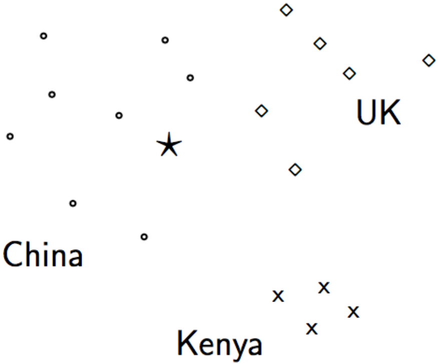
A word on Support Vector Machines...

- Each document is a vector, one component for each term.
- Terms are axes.
- High dimensionality: 10,000s of dimensions and more
- How can we do classification in this space?

A word on Support Vector Machines...

- As before, the training set is a set of documents, each labeled with its class.
- In vector space classification, this set corresponds to a labeled set of points or vectors in the vector space.
- Premise 1: Documents in the same class form a contiguous region.
- Premise 2: Documents from different classes don't overlap.
- We define lines, surfaces, hypersurfaces to divide regions.

A word on Support Vector Machines...

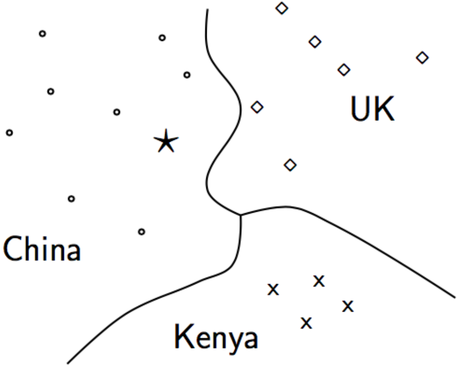


Classes in the vector space

Should the $\star$ document be assigned to Chine, UK or Kenya?

A word on Support Vector Machines...

Find separators between the classes



A word on Support Vector Machines...

Linear classifiers

- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i \beta_i x_i$ of the feature values.
  - Classification decision: $\sum_i \beta_i x_i > \beta_0$ (β_0 is our bias)
  - ... β_0, a parameter, is our classification threshold;
- We call this the **separator** or **decision boundary**.
- We find the separator based on training set.
- Methods for finding separator: logistic regression, linear SVM
- Assumption: The classes are **linearly separable**.

SVMs - geometric intuition

A Linear classifier in 1D

A linear classifier in 1D is a point X described by equation $\beta_1 x_1 = \beta_0$, where $x = \frac{\beta_0}{\beta_1}$; points (x_1) with $\beta_1 x_1 \geq \beta_0$ are in the class c;

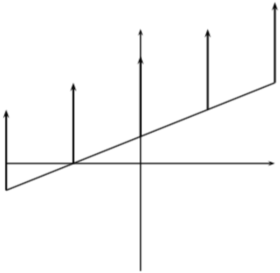
SVMs - geometric intuition

A Linear classifier in 1D

A linear classifier in 1D is a point X described by equation $\beta_1 x_1 = \beta_0$, where $x = \frac{\beta_0}{\beta_1}$; points (x_1) with $\beta_1 x_1 \geq \beta_0$ are in the class c; points with $\beta_1 x_1 \geq \beta_0$ are in the complement class $\hat{c}$

SVMs - geometric intuition

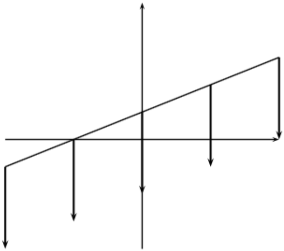
A Linear classifier in 2D



A linear classifier in 2D is a line described by equation $\beta_1 x_1 + \beta_2 x_2 = \beta_0$; points $(x_1 x_2)$ with $\beta_1 x_1 + \beta_2 x_2 \geq \beta_0$ are in the class c

SVMs - geometric intuition

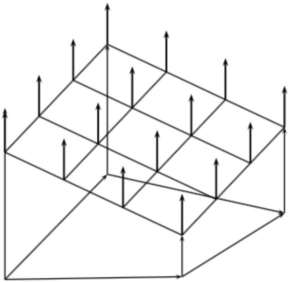
A Linear classifier in 2D



A linear classifier in 2D is a line described by equation $\beta_1 x_1 + \beta_2 x_2 = \beta_0$; points $(x_1 x_2)$ with $\beta_1 x_1 + \beta_2 x_2 \geq \beta_0$ are in the class c; points with $\beta_1 x_1 + \beta_2 x_2 \geq \beta_0$ are in the complement class $\hat{c}$

SVMs - geometric intuition

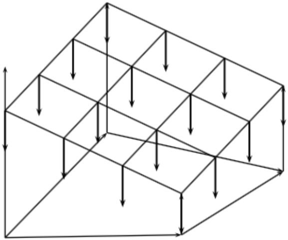
A Linear classifier in 3D



A linear classifier in 3D is a line described by equation
$\beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 = \beta_0$;

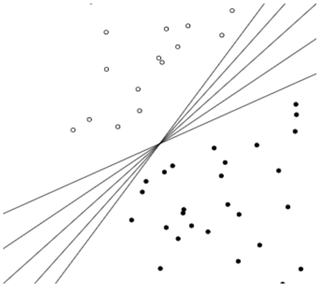
SVMs - geometric intuition

A Linear classifier in 3D



SVMs - definition

SVMs: A kind of large-margin classifier
Vector space based machine-learning method aiming to find a decision boundary between two classes that is maximally far from any point in the training data (possibly discounting some points as outliers or noise)



SVMs - definition

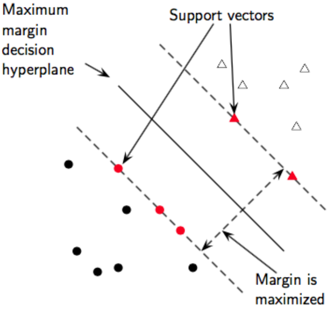
SVMs: A kind of large-margin classifier

2-class training data

decision boundary → **linear separator**

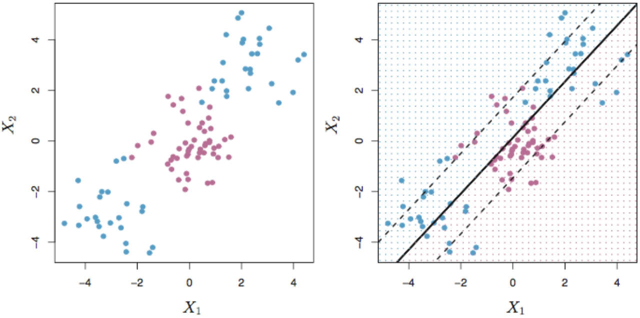
criterion: being maximally far away from any data point → determines classifier **margin**

linear separator position defined by **support vectors**



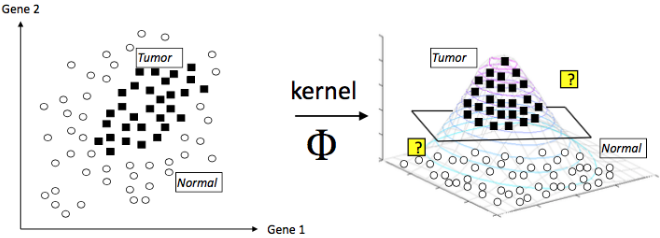
Why maximize the margin? It increaes ability to correctly generalize to test data;

What is there is no linear solution?



kernel trick...

SVMs represent the data in a higher dimensional projection using a kernel, and bisect this using a hyperplane Gene 2

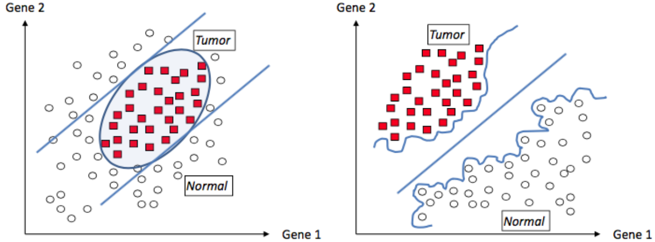


Data is not linearly separable in the input space

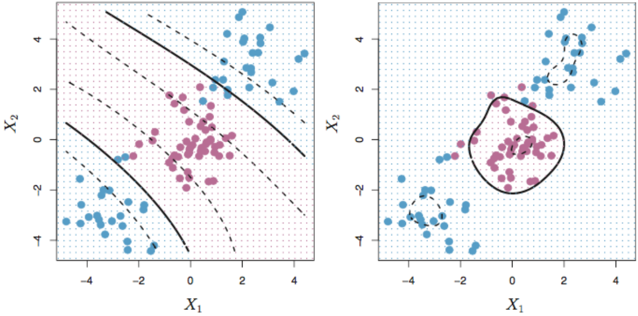
Data is linearly separable in the feature space obtained by a kernel

kernel trick...

This is only needed when no linear separation plane exists - so not needed in second of these



Kerlnels can give you different decision boundaries based on the different projections of data into higher-dimensional space



Ideological Scaling

1) Task
   - Measure political actors' position in policy space
   - Low dimensional representation of beliefs
2) Objective function
   - Linear Discriminant Analysis (ish)$\rightsquigarrow$ Wordscores
   - Item Response Theory $\rightsquigarrow$ Wordfish
   - Item Response Theory + Roll Call Votes $\rightsquigarrow$ Issue-specific ideal points
3) Optimization
   - Wordscores$\rightsquigarrow$ straightforward, based on training texts
   - Wordfish$\rightsquigarrow$ EM, MCMC methods
4) Validation
   - What is the goal of embedding?
   - What is the gold standard?

The Spatial Model

- Suppose we have actor i ($i = 1, 2, 3, \ldots, N$)

The Spatial Model

- Suppose we have actor i ($i = 1, 2, 3, \ldots, N$)
- Actor has ideal point $\boldsymbol{\theta}_i \in \Re^M$

The Spatial Model

- Suppose we have actor i ($i = 1, 2, 3, \ldots, N$)
- Actor has ideal point $\boldsymbol{\theta}_i \in \Re^M$
- We describe actor i's utility from proposal $\boldsymbol{p} \in \Re^M$ with utility function

The Spatial Model

- Suppose we have actor i ($i = 1, 2, 3, \ldots, N$)
- Actor has ideal point $\boldsymbol{\theta}_i \in \Re^M$
- We describe actor i's utility from proposal $\boldsymbol{p} \in \Re^M$ with utility function

$$u_i(\boldsymbol{\theta}_i, \boldsymbol{p}) = -d(\boldsymbol{\theta}_i, \boldsymbol{p})$$

The Spatial Model

- Suppose we have actor i ($i = 1, 2, 3, \ldots, N$)
- Actor has ideal point $\boldsymbol{\theta}_i \in \Re^M$
- We describe actor i's utility from proposal $\boldsymbol{p} \in \Re^M$ with utility function

$$
\begin{aligned}
u_i(\boldsymbol{\theta}_i, \boldsymbol{p}) &= -d(\boldsymbol{\theta}_i, \boldsymbol{p}) \\
&= -\sum_{l=1}^{L} (\underbrace{\theta_{il}}_{\text{ideal policy}} - p_l)^2
\end{aligned}
$$

The Spatial Model

- Suppose we have actor i ($i = 1, 2, 3, \ldots, N$)
- Actor has ideal point $\boldsymbol{\theta}_i \in \Re^M$
- We describe actor i's utility from proposal $\boldsymbol{p} \in \Re^M$ with utility function

$$
\begin{aligned}
u_i(\boldsymbol{\theta}_i, \boldsymbol{p}) &= -d(\boldsymbol{\theta}_i, \boldsymbol{p}) \\
&= -\sum_{l=1}^{L}(\underbrace{\theta_{il}}_{\text{ideal policy}} - p_l)^2
\end{aligned}
$$

Estimation goal: $\widehat{\boldsymbol{\theta}}_i$

The Spatial Model

- Suppose we have actor i ($i = 1, 2, 3, \ldots, N$)
- Actor has ideal point $\boldsymbol{\theta}_i \in \Re^M$
- We describe actor i's utility from proposal $\boldsymbol{p} \in \Re^M$ with utility function

$$\begin{aligned} u_i(\boldsymbol{\theta}_i, \boldsymbol{p}) &= -d(\boldsymbol{\theta}_i, \boldsymbol{p}) \\ &= -\sum_{l=1}^{L}(\underbrace{\theta_{il}}_{\text{ideal policy}} - p_l)^2 \end{aligned}$$

Estimation goal: $\widehat{\boldsymbol{\theta}}_i$
Scaling $\rightsquigarrow$ placing actors in low-dimensional space (like principal components!)

Estimating Ideal Points: Roll Call Data and the US Congress

US Congress and Roll Call

Estimating Ideal Points: Roll Call Data and the US Congress

US Congress and Roll Call

- Poole and Rosenthal `voteview`

Estimating Ideal Points: Roll Call Data and the US Congress

US Congress and Roll Call

- Poole and Rosenthal `voteview`
  - Roll Call Data $\rightsquigarrow$ 1789-Present

Estimating Ideal Points: Roll Call Data and the US Congress

US Congress and Roll Call

- Poole and Rosenthal `voteview`
  - Roll Call Data $\rightsquigarrow$ 1789-Present
  - `NOMINATE` methods $\rightsquigarrow$ place legislators on one dimension, estimate of ideology

Estimating Ideal Points: Roll Call Data and the US Congress

US Congress and Roll Call

- Poole and Rosenthal `voteview`
  - Roll Call Data $\rightsquigarrow$ 1789-Present
  - `NOMINATE` methods $\rightsquigarrow$ place legislators on one dimension, estimate of ideology
- Wildly successful:

Estimating Ideal Points: Roll Call Data and the US Congress

US Congress and Roll Call

- Poole and Rosenthal `voteview`
  - Roll Call Data$\rightsquigarrow$ 1789-Present
  - `NOMINATE` methods$\rightsquigarrow$ place legislators on one dimension, estimate of ideology
- Wildly successful:
  - Estimates are accurate: face validity Congressional scholars agree upon

Estimating Ideal Points: Roll Call Data and the US Congress

US Congress and Roll Call

- Poole and Rosenthal `voteview`
  - Roll Call Data $\rightsquigarrow$ 1789-Present
  - `NOMINATE` methods $\rightsquigarrow$ place legislators on one dimension, estimate of ideology
- Wildly successful:
  - Estimates are accurate: face validity Congressional scholars agree upon
  - Insightful $\rightsquigarrow$ unidimensional Congress

Estimating Ideal Points: Roll Call Data and the US Congress

US Congress and Roll Call

- Poole and Rosenthal `voteview`
  - Roll Call Data $\rightsquigarrow$ 1789-Present
  - `NOMINATE` methods $\rightsquigarrow$ place legislators on one dimension, estimate of ideology
- Wildly successful:
  - Estimates are accurate: face validity Congressional scholars agree upon
  - Insightful $\rightsquigarrow$ unidimensional Congress
  - Extensible: insight of IRT allows model to be embedded in many forms

Estimating Ideal Points: Roll Call Data and the US Congress

US Congress and Roll Call

- Poole and Rosenthal `voteview`
  - Roll Call Data$\rightsquigarrow$ 1789-Present
  - `NOMINATE` methods$\rightsquigarrow$ place legislators on one dimension, estimate of ideology
- Wildly successful:
  - Estimates are accurate: face validity Congressional scholars agree upon
  - Insightful$\rightsquigarrow$ unidimensional Congress
  - Extensible: insight of IRT allows model to be embedded in many forms
  - Widely used: hard to write a paper on American political institutions with ideal points

Estimating Ideal Points in General

Two Limitations with the NOMINATE project:

Estimating Ideal Points in General

Two Limitations with the NOMINATE project:

1) US Congress is distinct

Estimating Ideal Points in General

Two Limitations with the NOMINATE project:

1) US Congress is distinct $\rightsquigarrow$ roll call votes fail to measure ideology in other settings

Estimating Ideal Points in General

Two Limitations with the NOMINATE project:

1) US Congress is distinct $\rightsquigarrow$ roll call votes fail to measure ideology in other settings
   - Weak party pressure $\rightsquigarrow$ individual discretion on votes

Estimating Ideal Points in General

Two Limitations with the NOMINATE project:

1) US Congress is distinct$\rightsquigarrow$ roll call votes fail to measure ideology in other settings
   - Weak party pressure$\rightsquigarrow$ individual discretion on votes
   - Parliamentary systems$\rightsquigarrow$ no discretion, no variation.

Estimating Ideal Points in General

Two Limitations with the NOMINATE project:

1) US Congress is distinct $\rightsquigarrow$ roll call votes fail to measure ideology in other settings
   - Weak party pressure $\rightsquigarrow$ individual discretion on votes
   - Parliamentary systems $\rightsquigarrow$ no discretion, no variation.
   - Spirling and Quinn (2011) $\rightsquigarrow$ mixture model like models for blocs in UK Parliament

Estimating Ideal Points in General

Two Limitations with the NOMINATE project:

1) US Congress is distinct $\rightsquigarrow$ roll call votes fail to measure ideology in other settings
    - Weak party pressure $\rightsquigarrow$ individual discretion on votes
    - Parliamentary systems $\rightsquigarrow$ no discretion, no variation.
    - Spirling and Quinn (2011) $\rightsquigarrow$ mixture model like models for blocs in UK Parliament
2) Not everyone votes!

Estimating Ideal Points in General

Two Limitations with the NOMINATE project:

1) US Congress is distinct$\rightsquigarrow$ roll call votes fail to measure ideology in other settings
    - Weak party pressure$\rightsquigarrow$ individual discretion on votes
    - Parliamentary systems$\rightsquigarrow$ no discretion, no variation.
    - Spirling and Quinn (2011)$\rightsquigarrow$ mixture model like models for blocs in UK Parliament
2) Not everyone votes!
    - Voters$\rightsquigarrow$ survey responses (but problems with that)

Estimating Ideal Points in General

Two Limitations with the NOMINATE project:

1) US Congress is distinct $\rightsquigarrow$ roll call votes fail to measure ideology in other settings
   - Weak party pressure $\rightsquigarrow$ individual discretion on votes
   - Parliamentary systems $\rightsquigarrow$ no discretion, no variation.
   - Spirling and Quinn (2011) $\rightsquigarrow$ mixture model like models for blocs in UK Parliament
2) Not everyone votes!
   - Voters $\rightsquigarrow$ survey responses (but problems with that)
   - Challengers $\rightsquigarrow$ NPAT surveys (but they don't fill those out anymore)

Estimating Ideal Points in General

Two Limitations with the NOMINATE project:

1) US Congress is distinct $\rightsquigarrow$ roll call votes fail to measure ideology in other settings
   - Weak party pressure $\rightsquigarrow$ individual discretion on votes
   - Parliamentary systems $\rightsquigarrow$ no discretion, no variation.
   - Spirling and Quinn (2011) $\rightsquigarrow$ mixture model like models for blocs in UK Parliament
2) Not everyone votes!
   - Voters $\rightsquigarrow$ survey responses (but problems with that)
   - Challengers $\rightsquigarrow$ NPAT surveys (but they don't fill those out anymore)
   - Bonica (2013, 2014) $\rightsquigarrow$ estimate ideology from donations (but not everyone donates)

Estimating Ideal Points in General

But Everyone talks!

Estimating Ideal Points in General

But Everyone talks!

- If we could scale based on conversation, we can measure ideology anywhere

Estimating Ideal Points in General

But Everyone talks!

- If we could scale based on conversation, we can measure ideology anywhere
- Much of the literature $\rightsquigarrow$ relies upon intuition from US Congress

Estimating Ideal Points in General

But Everyone talks!

- If we could scale based on conversation, we can measure ideology anywhere
- Much of the literature$\rightsquigarrow$ relies upon intuition from US Congress
    - Hard not to find ideology

Estimating Ideal Points in General

But Everyone talks!

- If we could scale based on conversation, we can measure ideology anywhere
- Much of the literature$\rightsquigarrow$ relies upon intuition from US Congress
  - Hard not to find ideology
  - Behavior that is primarily ideological

Estimating Ideal Points in General

But Everyone talks!

- If we could scale based on conversation, we can measure ideology anywhere
- Much of the literature$\rightsquigarrow$ relies upon intuition from US Congress
  - Hard not to find ideology
  - Behavior that is primarily ideological
- Reality: scaling is much more difficult than roll call voting examples

Estimating Ideal Points in General

But Everyone talks!

- If we could scale based on conversation, we can measure ideology anywhere
- Much of the literature$\rightsquigarrow$ relies upon intuition from US Congress
  - Hard not to find ideology
  - Behavior that is primarily ideological
- Reality: scaling is much more difficult than roll call voting examples
  - Hard to find ideology

Estimating Ideal Points in General

But Everyone talks!

- If we could scale based on conversation, we can measure ideology anywhere
- Much of the literature $\rightsquigarrow$ relies upon intuition from US Congress
  - Hard not to find ideology
  - Behavior that is primarily ideological
- Reality: scaling is much more difficult than roll call voting examples
  - Hard to find ideology
  - Much of political speech reveals little about position on ideological spectrum $\rightsquigarrow$ advertising, regional

Estimating Ideal Points in General

But Everyone talks!

- If we could scale based on conversation, we can measure ideology anywhere
- Much of the literature $\rightsquigarrow$ relies upon intuition from US Congress
  - Hard not to find ideology
  - Behavior that is primarily ideological
- Reality: scaling is much more difficult than roll call voting examples
  - Hard to find ideology
  - Much of political speech reveals little about position on ideological spectrum $\rightsquigarrow$ advertising, regional

Healthy skepticism!

Wordscores (Laver, Benoit & Garry, 2003)

Long standing interest in scaling political texts relative to one another:

- e.g. are parties moving together over time, such that manifestos are converging?
- e.g. do members of parliament speak in line with their constituency's ideology (roll calls typically uninformative)?

→ LBG suggest a way of scoring documents in a NB style, so that we can answer such questions.

Basics

1 Begin with a reference set (training set) of texts that have known positions.

e.g. we find a 'left' document and give it score -1; and a 'right' document and give it score 1

2 Generate word scores from these reference texts

3 Score the virgin texts (test set) of texts using those word scores, possibly transform virgin scores to original metric.

Wordscores: Objective Function

For each legislator i, suppose we observe D_i documents.

Wordscores: Objective Function

For each legislator i, suppose we observe D_i documents.
Define:

Wordscores: Objective Function

For each legislator i, suppose we observe D_i documents.
Define:

$$\boldsymbol{x}_i \;=\; \sum_{l=1}^{D_i} \boldsymbol{x}_{il}$$

Wordscores: Objective Function

For each legislator i, suppose we observe D_i documents.
Define:

$$\begin{aligned} \boldsymbol{x}_i &= \sum_{l=1}^{D_i} \boldsymbol{x}_{il} \\ &= \sum_{l=1}^{D_i} (x_{il1}, x_{il2}, \ldots, x_{ilJ}) \end{aligned}$$

Wordscores: Objective Function

For each legislator i, suppose we observe D_i documents.
Define:

$$\begin{aligned} \boldsymbol{x}_i &= \sum_{l=1}^{D_i} \boldsymbol{x}_{il} \\ &= \sum_{l=1}^{D_i} (x_{il1}, x_{il2}, \ldots, x_{ilJ}) \end{aligned}$$

$\boldsymbol{x}_i \rightsquigarrow$ aggregation across documents, where each legislator is a row in the DTM (normalized by length speech)

Wordscores: Objective Functions

Choose two legislators as exemplars

Wordscores: Objective Functions

Choose two legislators as exemplars

- Legislator $L \in \{1, 2, \ldots, N\}$ is liberal. $Y_L = -1$

Wordscores: Objective Functions

Choose two legislators as exemplars

- Legislator $L \in \{1, 2, \ldots, N\}$ is liberal. $Y_L = -1$
- For example, might select Elizabeth Warren

Wordscores: Objective Functions

Choose two legislators as exemplars

- Legislator $L \in \{1, 2, \ldots, N\}$ is liberal. $Y_L = -1$
- For example, might select Elizabeth Warren
- Legislator $C \in \{1, 2, \ldots, N\}$ is Conservative. $Y_C = 1$

Wordscores: Objective Functions

Choose two legislators as exemplars

- Legislator $L \in \{1, 2, \ldots, N\}$ is liberal. $Y_L = -1$
- For example, might select Elizabeth Warren
- Legislator $C \in \{1, 2, \ldots, N\}$ is Conservative. $Y_C = 1$
- For example, might select Ted Cruz

Wordscores: Objective Functions

Choose two legislators as exemplars

- Legislator $L \in \{1, 2, \ldots, N\}$ is liberal. $Y_L = -1$
- For example, might select Elizabeth Warren
- Legislator $C \in \{1, 2, \ldots, N\}$ is Conservative. $Y_C = 1$
- For example, might select Ted Cruz

For each word j we can define:

Wordscores: Objective Functions

Choose two legislators as exemplars

- Legislator $L \in \{1, 2, \ldots, N\}$ is liberal. $Y_L = -1$
- For example, might select Elizabeth Warren
- Legislator $C \in \{1, 2, \ldots, N\}$ is Conservative. $Y_C = 1$
- For example, might select Ted Cruz

For each word j we can define:

$$P_{jL} = \text{Probability of word from Liberal}$$

Wordscores: Objective Functions

Choose two legislators as exemplars

- Legislator $L \in \{1, 2, \ldots, N\}$ is liberal. $Y_L = -1$
- For example, might select Elizabeth Warren
- Legislator $C \in \{1, 2, \ldots, N\}$ is Conservative. $Y_C = 1$
- For example, might select Ted Cruz

For each word j we can define:

$$\begin{aligned} P_{jL} &= \text{Probability of word from Liberal} \\ P_{jC} &= \text{Probability of word from Conservative} \end{aligned}$$

Wordscores: Objective Functions

Choose two legislators as exemplars

- Legislator $L \in \{1, 2, \ldots, N\}$ is liberal. $Y_L = -1$
- For example, might select Elizabeth Warren
- Legislator $C \in \{1, 2, \ldots, N\}$ is Conservative. $Y_C = 1$
- For example, might select Ted Cruz

For each word j we can define:

$$
\begin{aligned}
P_{jL} &= \text{Probability of word from Liberal} \\
P_{jC} &= \text{Probability of word from Conservative}
\end{aligned}
$$

Define the score for word j

Wordscores: Objective Functions

Choose two legislators as exemplars

- Legislator $L \in \{1, 2, \ldots, N\}$ is liberal. $Y_L = -1$
- For example, might select Elizabeth Warren
- Legislator $C \in \{1, 2, \ldots, N\}$ is Conservative. $Y_C = 1$
- For example, might select Ted Cruz

For each word j we can define:

$$\begin{aligned} P_{jL} &= \text{Probability of word from Liberal} \\ P_{jC} &= \text{Probability of word from Conservative} \end{aligned}$$

Define the score for word j

$$S_j = Y_C P_{jC} + Y_L P_{jL}$$

Wordscores: Objective Functions

Choose two legislators as exemplars

- Legislator $L \in \{1, 2, \ldots, N\}$ is liberal. $Y_L = -1$
- For example, might select Elizabeth Warren
- Legislator $C \in \{1, 2, \ldots, N\}$ is Conservative. $Y_C = 1$
- For example, might select Ted Cruz

For each word j we can define:

$$\begin{aligned} P_{jL} &= \text{Probability of word from Liberal} \\ P_{jC} &= \text{Probability of word from Conservative} \end{aligned}$$

Define the score for word j

$$\begin{aligned} S_j &= Y_C P_{jC} + Y_L P_{jL} \\ &= P_{jC} - P_{jL} \end{aligned}$$

Wordscores: Objective Functions

Scale other legislators. First let's count the number of words they've spoken:

Wordscores is essentially estimating a dictionary. The more negative their speech score is, the closer they get to the Liberal position. Inverse is true for conservative.

Wordscores: Objective Functions

Scale other legislators. First let's count the number of words they've spoken:

$$N_i \;=\; \sum_{j=1}^{J} \boldsymbol{x}_j$$

Wordscores is essentially estimating a dictionary. The more negative their speech score is, the closer they get to the Liberal position. Inverse is true for conservative.

Wordscores: Objective Functions

Scale other legislators. First let's count the number of words they've spoken:

$$N_i = \sum_{j=1}^{J} \boldsymbol{x}_j$$

$\hat{\theta}_i$ is the sum over all the words for the rate at which the individual legislator uses word i, times the score S_j

Wordscores is essentially estimating a dictionary. The more negative their speech score is, the closer they get to the Liberal position. Inverse is true for conservative.

Wordscores: Objective Functions

Scale other legislators. First let's count the number of words they've spoken:

$$N_i = \sum_{j=1}^{J} \boldsymbol{x}_j$$

$\hat{\theta}_i$ is the sum over all the words for the rate at which the individual legislator uses word i, times the score S_j

$$\hat{\theta}_i = \sum_{j=1}^{J} \left(\frac{x_{ij}}{N_i} \right) S_j$$

Wordscores is essentially estimating a dictionary. The more negative their speech score is, the closer they get to the Liberal position. Inverse is true for conservative.

Wordscores: Objective Functions

Scale other legislators. First let's count the number of words they've spoken:

$$N_i = \sum_{j=1}^{J} \boldsymbol{x}_j$$

$\hat{\theta}_i$ is the sum over all the words for the rate at which the individual legislator uses word i, times the score S_j

$$\begin{aligned} \hat{\theta}_i &= \sum_{j=1}^{J} \left(\frac{x_{ij}}{N_i} \right) S_j \\ &= \frac{\boldsymbol{x}_i^{'}}{N_i} \boldsymbol{S} \end{aligned}$$

Wordscores is essentially estimating a dictionary. The more negative their speech score is, the closer they get to the Liberal position. Inverse is true for conservative.

Wordscores: Optimization

Let's count the number of words from Lib and Con:

$$
\begin{aligned}
N_L &= \sum_{m=1}^{J} x_{mL} \\
N_C &= \sum_{m=1}^{J} x_{mC}
\end{aligned}
$$

Wordscores: Optimization

Let's count the number of words from Lib and Con:

$$N_L = \sum_{m=1}^{J} x_{mL}$$

$$N_C = \sum_{m=1}^{J} x_{mC}$$

Estimate P_{jL}, P_{jC}, and S_j

Wordscores: Optimization

Let's count the number of words from Lib and Con:

$$N_L = \sum_{m=1}^{J} x_{mL}$$
$$N_C = \sum_{m=1}^{J} x_{mC}$$

Estimate P_{jL}, P_{jC}, and S_j

Thisissimplytherateatwhichtheliberalusesthewordoverthetotalratethatthewordis

Wordscores: Optimization

Let's count the number of words from Lib and Con:

$$N_L = \sum_{m=1}^{J} x_{mL}$$

$$N_C = \sum_{m=1}^{J} x_{mC}$$

Estimate P_{jL}, P_{jC}, and S_j

$$P_{jL} = \frac{\frac{x_{jL}}{N_L}}{\frac{x_{jL}}{N_L} + \frac{x_{jC}}{N_C}}$$

$$P_{jC} = 1 - P_{jL} = \frac{\frac{x_{jC}}{N_C}}{\frac{x_{jL}}{N_L} + \frac{x_{jC}}{N_C}}$$

Wordscores: Optimization

Let's count the number of words from Lib and Con:

$$N_L = \sum_{m=1}^{J} x_{mL}$$
$$N_C = \sum_{m=1}^{J} x_{mC}$$

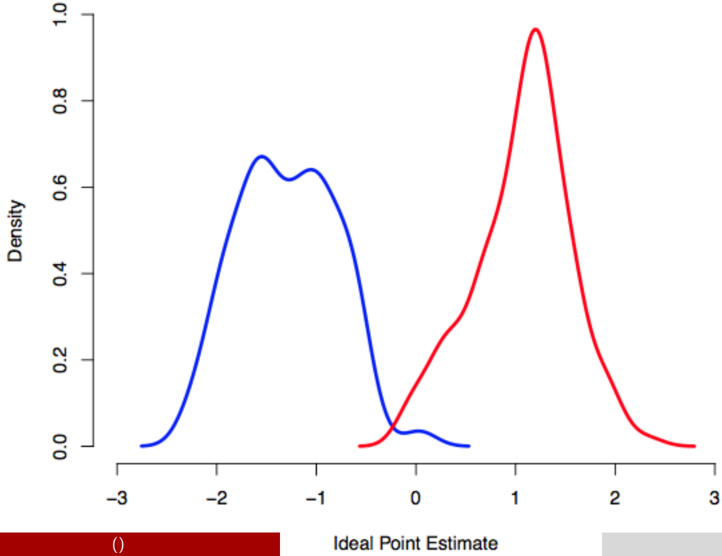
Estimate P_{jL}, P_{jC}, and S_j

$$P_{jL} = \frac{\frac{x_{jL}}{N_L}}{\frac{x_{jL}}{N_L} + \frac{x_{jC}}{N_C}}$$
$$P_{jC} = 1 - P_{jL} = \frac{\frac{x_{jC}}{N_C}}{\frac{x_{jL}}{N_L} + \frac{x_{jC}}{N_C}}$$
$$S_j = P_{jC} - P_{jL}$$

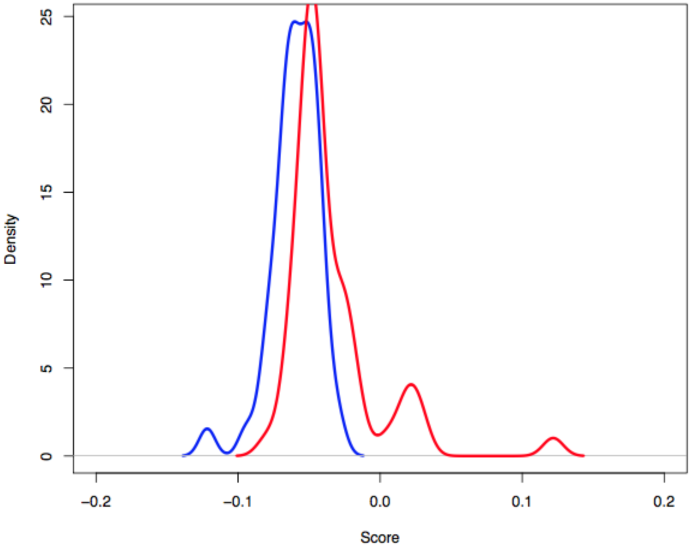
Applied to the Senate Press Releases

$L =$ Ted Kennedy
$C =$ Tom Coburn
Apply to other senators.

Applying to Senate Press Releases$\rightsquigarrow$ Gold Standard Scaling from NOMINATE



Applying to Senate Press Releases⇝ WordScores



Example

Neo-Nazi manifesto uses '`immigrant`' 25 times in 1000 words, while Communists use it only 5 times.

then $P_{iR} = \frac{0.025}{0.025+0.005} = 0.83.$

and $P_{iL} = \frac{0.005}{0.025+0.005} = 0.16.$

so $S_i = 0.83 - 0.16 = 0.66$

we see a virgin manifesto, from the Conservative party, and it mentions `immigrant` 20 times in a thousand words.

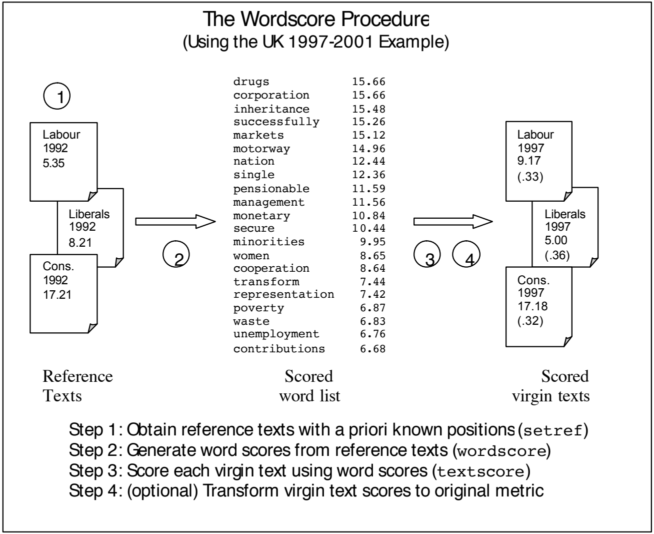
well the relevant calculation for that word is $0.02 \times 0.66 = 0.0132$.

but virgin manifesto, from Labour party, mentions it 10 times in a thousand words: $0.01 \times 0.66 = 0.006$

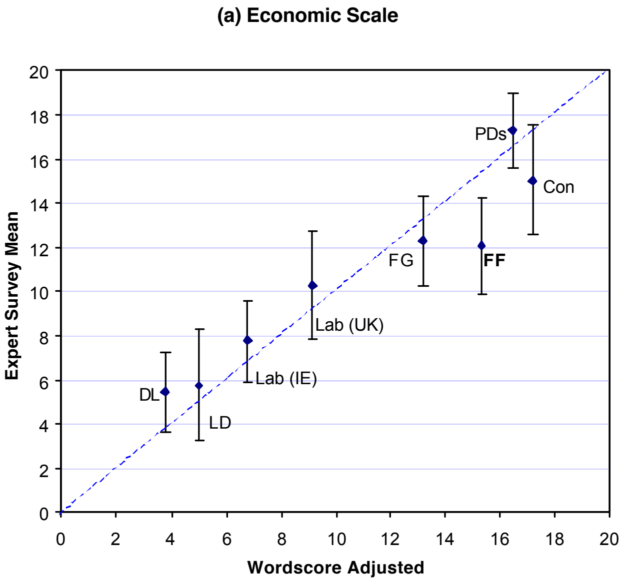
$\rightarrow$ can rescale these back to original $(-1, 1)$ dimension.

New Labour Moderates its Economic Policy

New Labour Moderates its Economic Policy



Compared to Expert Surveys



Comments

Extremely influential approach: avoids having to pick features of interest (features that don't distinguish between reference texts have $S_i = 0$)

and helpful/valid in practice, and can have uncertainty estimates to boot.

very important to obtain extreme and appropriate reference, and score them appropriately. Need to be from domain of virgin texts, and have lots of words.

but Lowe (typically?) unhappy (2008): no statistical model, inconsistent scoring assumptions, and difficult to pick up 'centrist language' (is equivalent to any language used commonly by all parties for linguistic reasons).

while Beauchamp (2011) provides comparison and extension to more purely Bayesian approach.

Cross-Validation: Some Intuition

Recall Optimal division of data:

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

K-fold Cross-validation idea: create many training and test sets.

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets
- Each step: use held out data to evaluate performance

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets
- Each step: use held out data to evaluate performance
- Avoid overfitting and have context specific penalty

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets
- Each step: use held out data to evaluate performance
- Avoid overfitting and have context specific penalty

Estimates:

$$\text{Error} \quad = \quad \mathrm{E}\left[\mathrm{E}[L(\boldsymbol{Y}, f(\hat{\boldsymbol{\beta}}, \boldsymbol{X}, \boldsymbol{\lambda}))|\mathcal{T}]\right]$$

Process:

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, . . ., Group K)

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, . . ., Group K)
- Rotate through groups as follows

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, . . ., Group K)
- Rotate through groups as follows

Step	Training	Validation ("Test")

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, . . ., Group K)
- Rotate through groups as follows

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, . . ., Group K	Group 1

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, . . ., Group K)
- Rotate through groups as follows

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, . . ., Group K	Group 1
2	Group 1, Group3, Group 4, . . ., Group K	Group 2

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, . . ., Group K)
- Rotate through groups as follows

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, . . ., Group K	Group 1
2	Group 1, Group3, Group 4, . . ., Group K	Group 2
3	Group 1, Group 2, Group 4, . . ., Group K	Group 3

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, ..., Group K)
- Rotate through groups as follows

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, . . ., Group K)
- Rotate through groups as follows

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, . . ., Group K	Group 1
2	Group 1, Group3, Group 4, . . ., Group K	Group 2
3	Group 1, Group 2, Group 4, . . ., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, . . ., Group K - 1	Group K

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, . . ., Group K	Group 1
2	Group 1, Group3, Group 4, . . ., Group K	Group 2
3	Group 1, Group 2, Group 4, . . ., Group K	Group 3
$\vdots$	$\vdots$	$\vdots$
K	Group 1, Group 2, Group 3, . . ., Group K - 1	Group K

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, . . ., Group K	Group 1
2	Group 1, Group3, Group 4, . . ., Group K	Group 2
3	Group 1, Group 2, Group 4, . . ., Group K	Group 3
$\vdots$	$\vdots$	$\vdots$
K	Group 1, Group 2, Group 3, . . ., Group K - 1	Group K

Strategy:

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
$\vdots$	$\vdots$	$\vdots$
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
$\vdots$	$\vdots$	$\vdots$
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K - 1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{X}, \boldsymbol{\lambda})$

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
$\vdots$	$\vdots$	$\vdots$
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K-1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{X}, \boldsymbol{\lambda})$
- Predict values for K^{th}

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K - 1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{X}, \boldsymbol{\lambda})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{X}, \boldsymbol{\lambda}))$

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
$\vdots$	$\vdots$	$\vdots$
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K-1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{X}, \boldsymbol{\lambda})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{X}, \boldsymbol{\lambda}))$
  - Mean square error, Absolute error, Prediction error, ...

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
$\vdots$	$\vdots$	$\vdots$
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K-1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{X}, \boldsymbol{\lambda})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{X}, \boldsymbol{\lambda}))$
    - Mean square error, Absolute error, Prediction error, ...

CV(ind. classification) $= \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{Y}_i, f^{-k}(\boldsymbol{X}_i))$

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
$\vdots$	$\vdots$	$\vdots$
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K - 1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{X}, \boldsymbol{\lambda})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{X}, \boldsymbol{\lambda}))$
    - Mean square error, Absolute error, Prediction error, ...

  CV(ind. classification) $= \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{Y}_i, f^{-k}(\boldsymbol{X}_i))$

  CV(proportions) $=$
  $\frac{1}{K}\sum_{j=1}^{K}$ Mean Square Error Proportions from Group j

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
$\vdots$	$\vdots$	$\vdots$
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

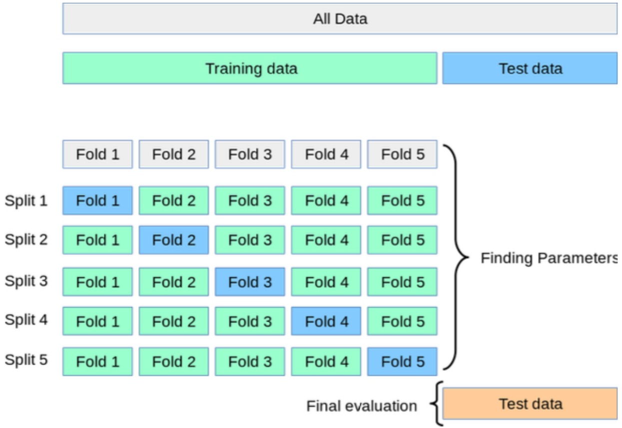
Strategy:

- Divide data into K groups
- Train data on $K-1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{X}, \boldsymbol{\lambda})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{X}, \boldsymbol{\lambda}))$
  - Mean square error, Absolute error, Prediction error, ...

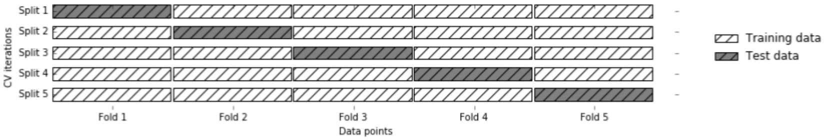
  CV(ind. classification) $= \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{Y}_i, f^{-k}(\boldsymbol{X}_i))$

  CV(proportions) $=$
  $\frac{1}{K}\sum_{j=1}^{K}$ Mean Square Error Proportions from Group j
- Final choice: model with highest CV score

visual intuition. . .



visual intuition...



pro: more stable, more data

con: slower

How Do We Select K?

Common values of K

- $K = 5$: Five fold cross validation
- $K = 10$: Ten fold cross validation
- $K = N$: Leave one out cross validation

Considerations:

- How sensitive are inferences to number of coded documents?
- 200 labeled documents
  - $K = N \rightarrow 199$ documents to train,
  - $K = 10 \rightarrow 180$ documents to train
  - $K = 5 \rightarrow 160$ documents to train
- 50 labeled documents
  - $K = N \rightarrow 49$ documents to train,
  - $K = 10 \rightarrow 45$ documents to train
  - $K = 5 \rightarrow 40$ documents to train
- How long will it take to run models?
  - $K-$fold cross validation requires $K\times$ One model run
- What is the correct loss function?