Text as Data

Justin Grimmer

Associate Professor
Department of Political Science
University of Chicago

August 17th, 2017

Discovery and Measurement

What is the research process? (Grimmer, Roberts, and Stewart 2017)

1) Discovery: a hypothesis or view of the world
2) Measurement according to some organization
3) Causal Inference: effect of some intervention

Text as data methods assist at each stage of research process

Measurement

Two approaches to measurement

1) Use an existing classification scheme to categorize documents (Today and Tuesday)
2) Simultaneously discover categories and measure prevalence (repurpose discovery methods) (Wednesday)

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 and 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.

Pre-existing word weights $\rightsquigarrow$ Dictionaries

`DICTION`

DICTION 7, now with *Power Mode,* can read a variety of text formats and can accept a large number o
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$\rightsquigarrow$ Dictionaries

`DICTION`

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

Pre-existing word weights⇝ Dictionaries

DICTION

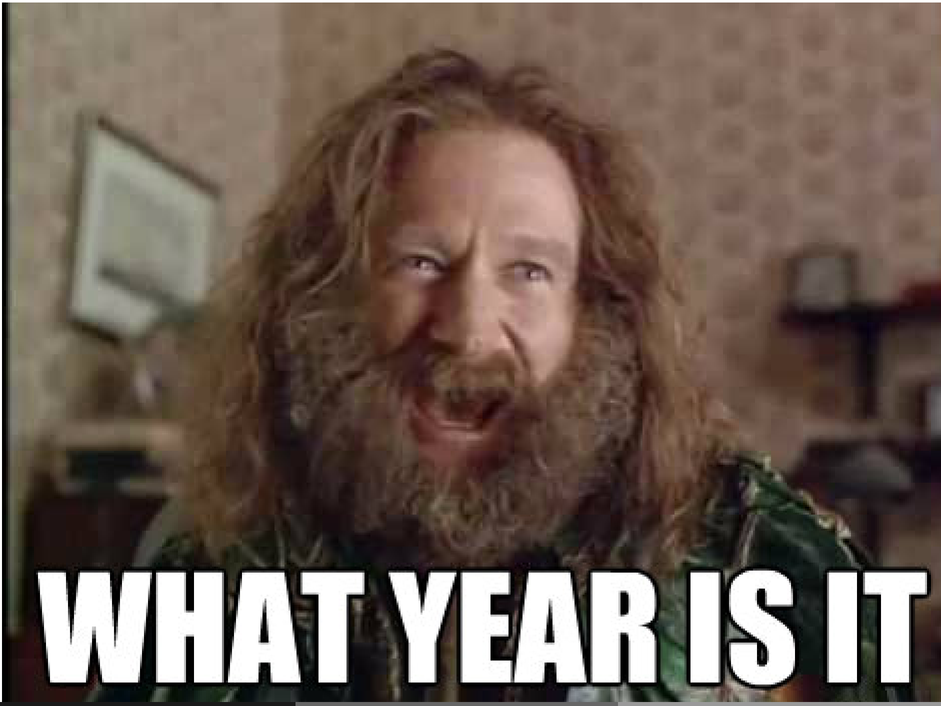
> “provides both social scientific and humanistic understandings"
> —Don Waisanen, Baruch College

Pre-existing word weights$\rightsquigarrow$ 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)

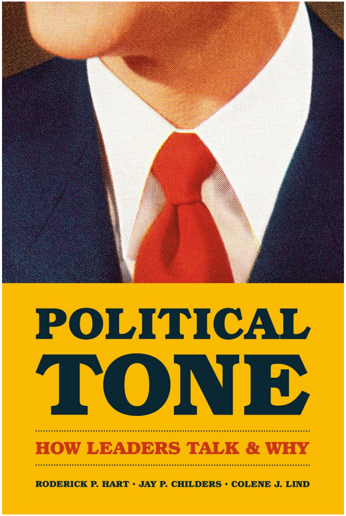
1) Proprietary$\rightsquigarrow$ wrapped in GUI
2) Basic tasks:
   a) Count words
   b) Weighted counts of words
   c) Some graphics

Dictionary Methods

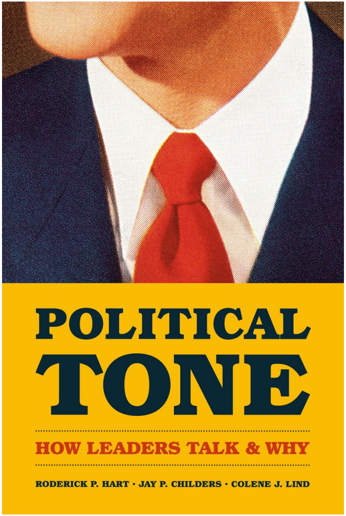
Many Dictionary Methods (like DICTION)

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

DICTION

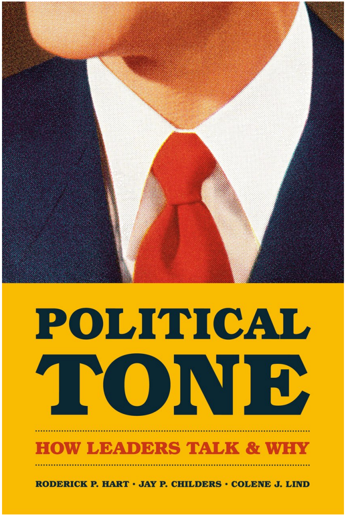

POLITICAL
TONE
HOW LEADERS TALK & WHY
RODERICK P. HART · JAY P. CHILDERS · COLENE J. LIND

DICTION



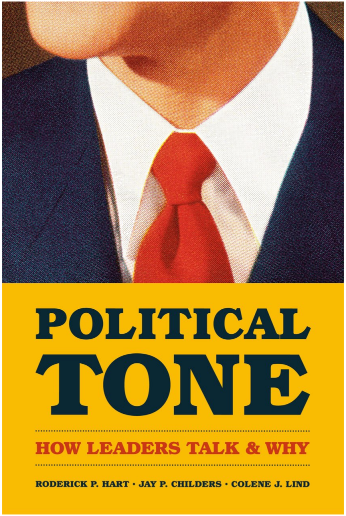
- { Certain, Uncertain }

DICTION



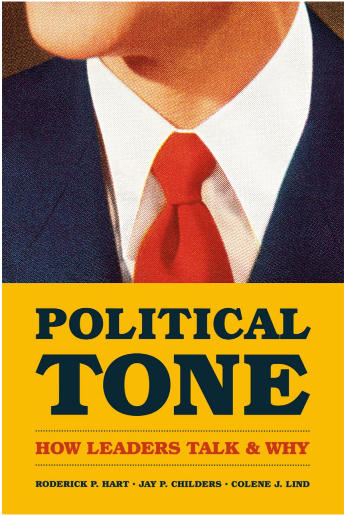
- { Certain, Uncertain }
  , { Optimistic, Pessimistic }

DICTION



- { Certain, Uncertain }
  , { Optimistic, Pessimistic }
- $\approx$ 10,000 words

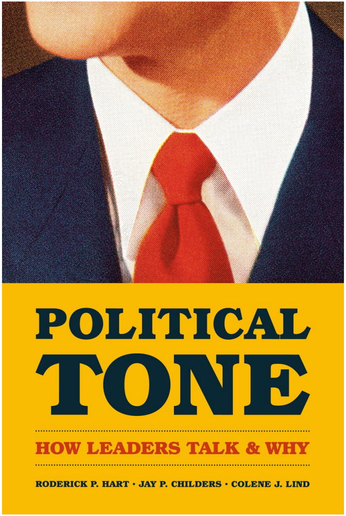
DICTION



- { Certain, Uncertain } , { Optimistic, Pessimistic }
- $\approx$ 10,000 words

Applies DICTION to a wide array of political texts

DICTION



- { Certain, Uncertain }
  , { Optimistic, Pessimistic }
- $\approx$ 10,000 words

Applies DICTION to a wide array of political texts
Examine specific periods of American political history

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⇝ " 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⇝ (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$\rightsquigarrow$ " 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

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 $\rightsquigarrow$ " 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)

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)
- ...

Generating New Words

Three ways to create dictionaries (non-exhaustive):

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)
- Manual generation

Generating New Words

Three ways to create dictionaries (non-exhaustive):

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

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating 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 (Separating 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 (Separating 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 (Separating 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 (Separating 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

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating 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 (Separating 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

Applying Methods to Documents

Applying the model:

Applying Methods to Documents

Applying the model:

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

Applying Methods to Documents

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 $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$

Applying Methods to Documents

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 $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$

Applying Methods to Documents

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 $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$

Applying Methods to Documents

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 $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
  - $\theta_k \in \{0, 1\}$
  - $\theta_k \in \{-1, 0, 1\}$
  - $\theta_k \in \{-2, -1, 0, 1, 2\}$

Applying Methods to Documents

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 $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

Applying Methods to Documents

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

Applying Methods to Documents

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

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

Applying Methods to Documents

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

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

Applying Methods to Documents

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

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

$$Y_i = \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \mathbf{1}}$$

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

Applying Methods to Documents

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

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

$$Y_i = \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \boldsymbol{1}}$$

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

$Y_i > 0 \Rightarrow$ Positive Category

Applying Methods to Documents

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

Applying Methods to Documents

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

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

$$Y_i = \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \boldsymbol{1}}$$

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

$Y_i > 0 \Rightarrow$ Positive Category

$Y_i < 0 \Rightarrow$ Negative Category

$Y_i \approx 0$ Ambiguous

Applying a Dictionary to Press Releases

Applying a Dictionary to Press Releases

- Collection of 169,779 press releases (US House members 2005-2010)

Applying a Dictionary to Press Releases

- Collection of 169,779 press releases (US House members 2005-2010)
- Dictionary from Neal Caren's website $\rightsquigarrow$ Theresa Wilson, Janyce Wiebe, and Paul Hoffman's dictionary

Applying a Dictionary to Press Releases

- Collection of 169,779 press releases (US House members 2005-2010)
- Dictionary from Neal Caren's website $\rightsquigarrow$ Theresa Wilson, Janyce Wiebe, and Paul Hoffman's dictionary
- Create positive/negative score for press releases.

Applying a Dictionary to Press Releases

- Collection of 169,779 press releases (US House members 2005-2010)
- Dictionary from Neal Caren's website $\rightsquigarrow$ Theresa Wilson, Janyce Wiebe, and Paul Hoffman's dictionary
- Create positive/negative score for press releases.

`Python` code and press releases

Examining Positive and Negative Statements in Press Releases

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007
4) John Boehner, 2009

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007
4) John Boehner, 2009
5) Jeff Flake, (basically all years)

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007
4) John Boehner, 2009
5) Jeff Flake, (basically all years)
6) Eric Cantor, 2009

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007
4) John Boehner, 2009
5) Jeff Flake, (basically all years)
6) Eric Cantor, 2009
7) Tom Price, 2010

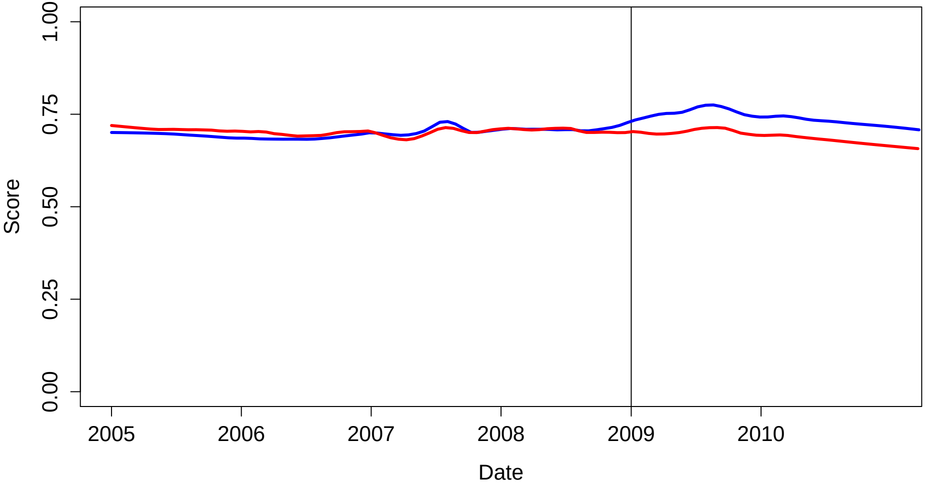
Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007
4) John Boehner, 2009
5) Jeff Flake, (basically all years)
6) Eric Cantor, 2009
7) Tom Price, 2010

Legislators who are more extreme $\rightsquigarrow$ less positive in press releases

Examining Positive and Negative Statements in Press Releases



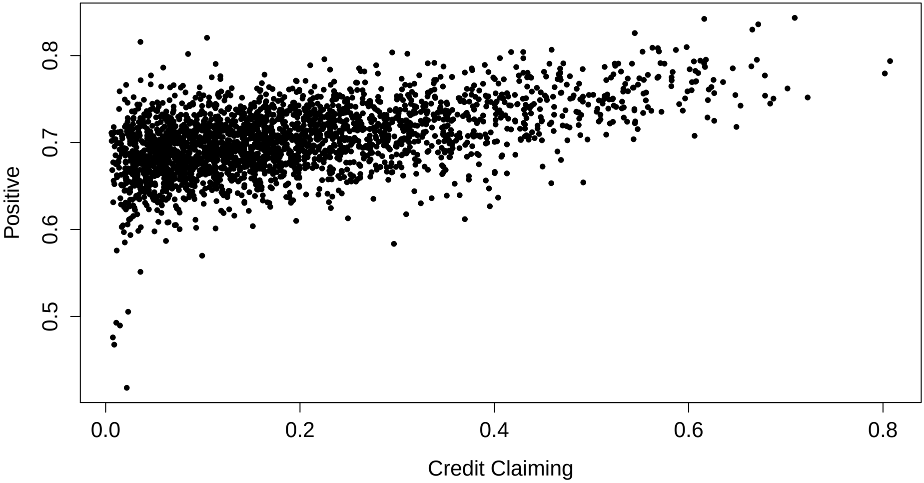
Examining Positive and Negative Statements in Press Releases

- Credit Claiming press release: 9.1 percentage points “more positive” than a non-credit claiming press release

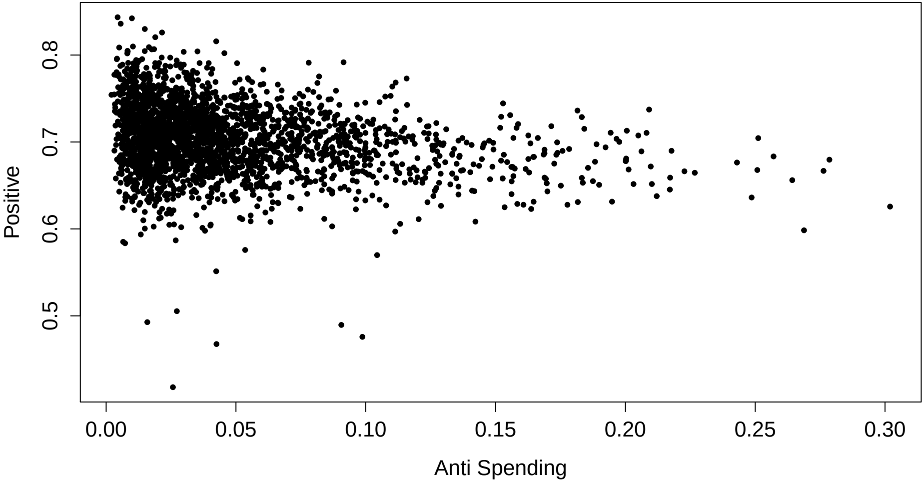
Examining Positive and Negative Statements in Press Releases

- Credit Claiming press release: 9.1 percentage points "more positive" than a non-credit claiming press release
- Anti-spending press release: 10.6 percentage points "less positive" than a non-anti spending press release

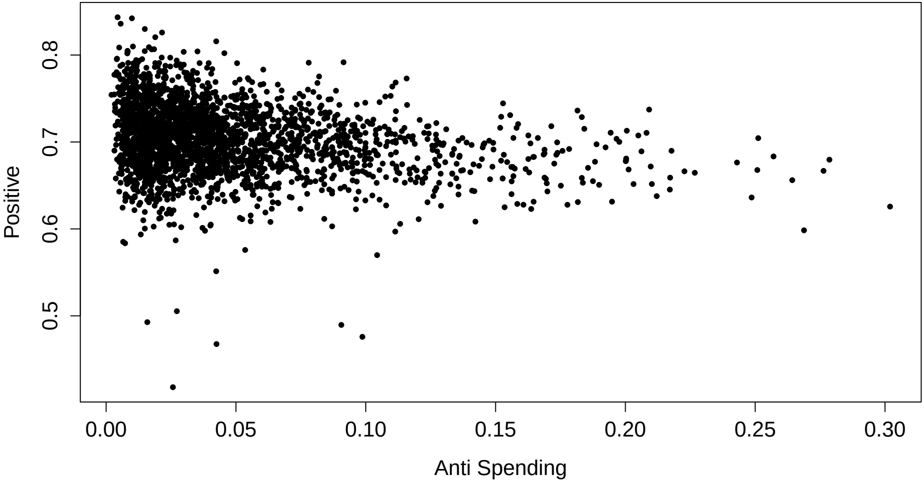
Examining Positive and Negative Statements in Press Releases



Examining Positive and Negative Statements in Press Releases



Examining Positive and Negative Statements in Press Releases



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

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

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

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

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise

$\rightsquigarrow$

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)

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

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

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

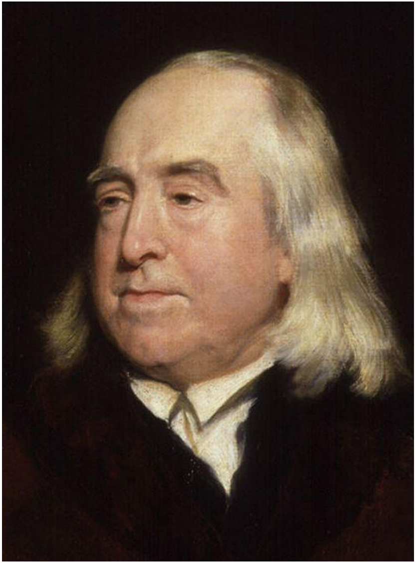
- tone matters ($)

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

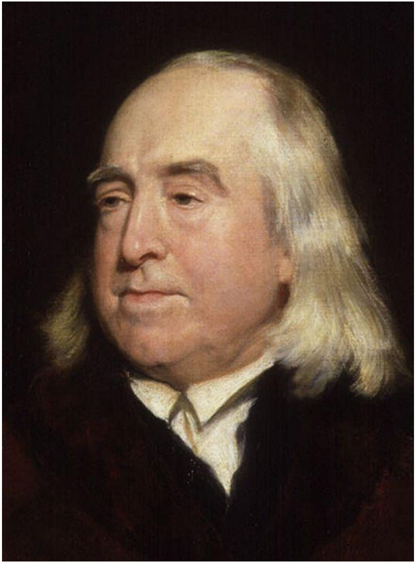
Loughran and McDonald (2011): Financial Documents are Different, polysemes

- 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, andunanticipated`

Measuring Happiness

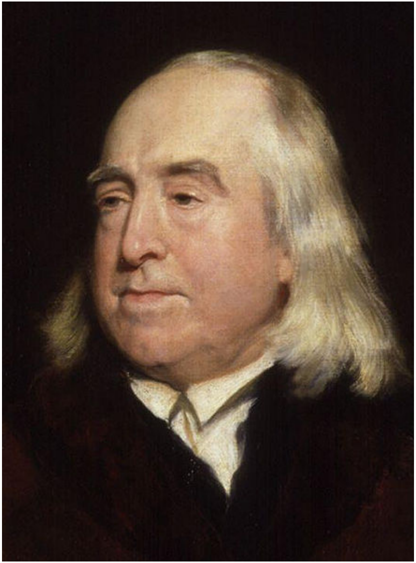


Measuring Happiness



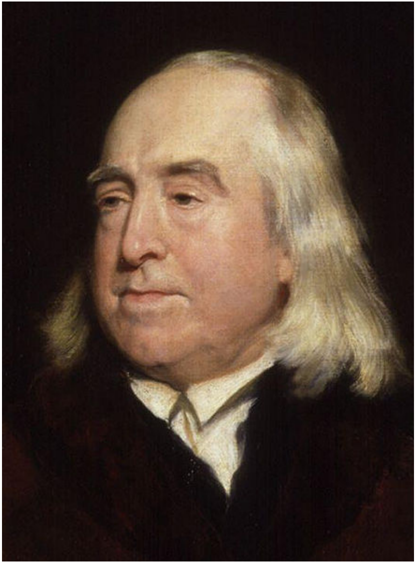
- Quantifying Happiness: How happy is society?

Measuring Happiness



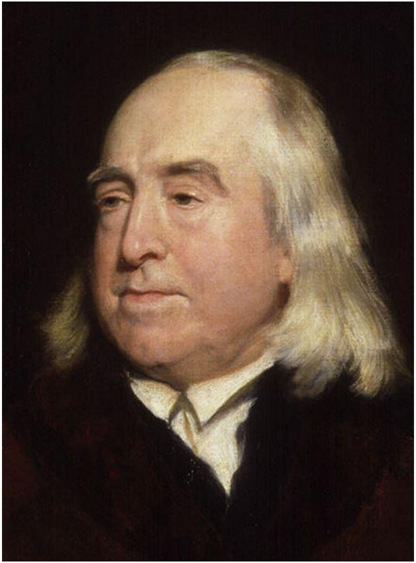
- Quantifying Happiness: How happy is society?
- How Happy is a Song?

Measuring Happiness



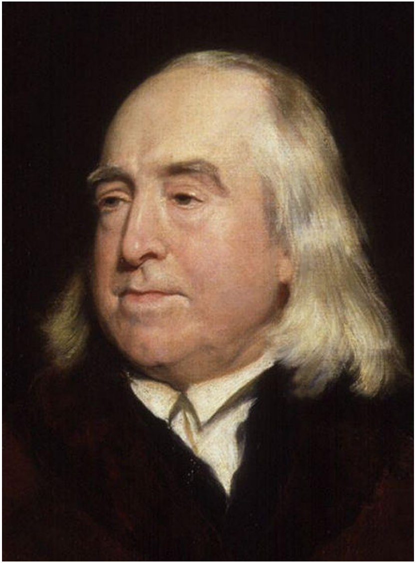
- Quantifying Happiness: How happy is society?
- How Happy is a Song?
- Blog posts?

Measuring Happiness



- Quantifying Happiness: How happy is society?
- How Happy is a Song?
- Blog posts?
- Facebook posts? (Gross National Happiness)

Measuring Happiness



- Quantifying Happiness: How happy is society?
- How Happy is a Song?
- Blog posts?
- Facebook posts? (Gross National Happiness)

Use Dictionary Methods

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
  - On a scale of 1-9 how happy does this word make you?

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
  - On a scale of 1-9 how happy does this word make you?
    Happy : triumphant (8.82)/paradise (8.72)/ love (8.72)

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
  - On a scale of 1-9 how happy does this word make you?
    Happy : triumphant (8.82)/paradise (8.72)/ love (8.72)
    Neutral: street (5.22)/ paper (5.20)/ engine (5.20)

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
  - On a scale of 1-9 how happy does this word make you?
    Happy : triumphant (8.82)/paradise (8.72)/ love (8.72)
    Neutral: street (5.22)/ paper (5.20)/ engine (5.20)
    Unhappy : cancer (1.5)/funeral (1.39)/ rape (1.25) /suicide (1.25)

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
  - On a scale of 1-9 how happy does this word make you?
    Happy : triumphant (8.82)/paradise (8.72)/ love (8.72)
    Neutral: street (5.22)/ paper (5.20)/ engine (5.20)
    Unhappy : cancer (1.5)/funeral (1.39)/ rape (1.25) /suicide (1.25)
- Happiness for text i (with word j having happiness θ_j and document frequence X_{ij}

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
    - On a scale of 1-9 how happy does this word make you?
      Happy : triumphant (8.82)/paradise (8.72)/ love (8.72)
      Neutral: street (5.22)/ paper (5.20)/ engine (5.20)
      Unhappy : cancer (1.5)/funeral (1.39)/ rape (1.25) /suicide (1.25)
- Happiness for text i (with word j having happiness θ_j and document frequence X_{ij}

$$\text{Happiness}_i = \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_{ik}}$$

Lyrics for
Michael Jackson's Billie Jean

"She was more like a beauty queen
from a movie scene.
⋮
And mother always told me,
be careful who you love.
And be careful of what you do
'cause the lie becomes the truth.
Billie Jean is not my lover,
She's just a girl who claims
that I am the one.
⋮

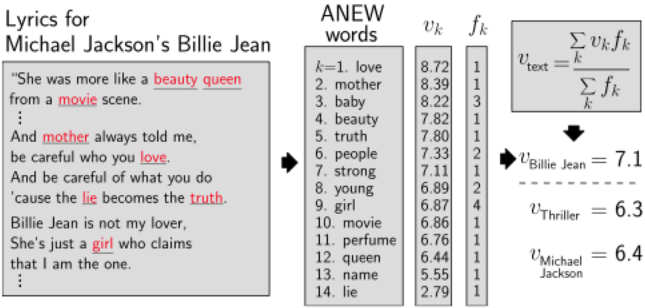
ANEW words	v_k	f_k
k=1. love	8.72	1
2. mother	8.39	1
3. baby	8.22	3
4. beauty	7.82	1
5. truth	7.80	1
6. people	7.33	2
7. strong	7.11	1
8. young	6.89	2
9. girl	6.87	4
10. movie	6.86	1
11. perfume	6.76	1
12. queen	6.44	1
13. name	5.55	1
14. lie	2.79	1

$$v_{\text{text}} = \frac{\sum_k v_k f_k}{\sum_k f_k}$$

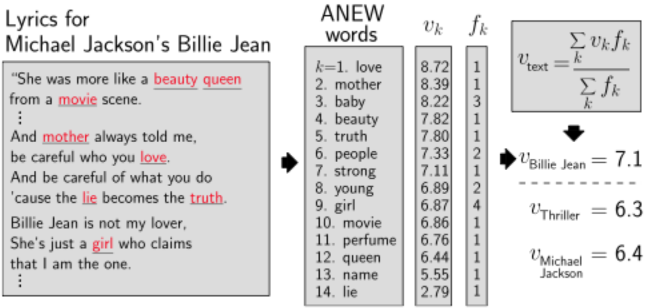
$$v_{\text{Billie Jean}} = 7.1$$

$$v_{\text{Thriller}} = 6.3$$

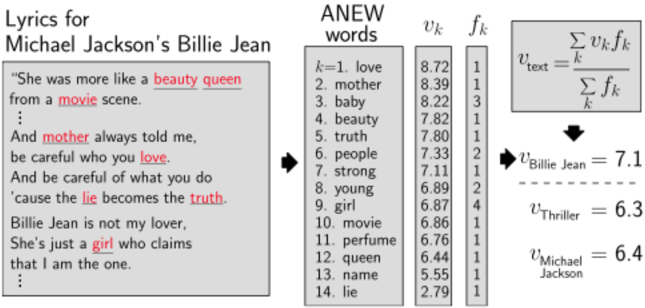
$$v_{\text{Michael Jackson}} = 6.4$$



Homework Hints: One approach: write a `for` loop searching for words in dictionary (caution: is dictionary stemmed?)



Homework Hints: One approach: write a `for` loop searching for words in dictionary (caution: is dictionary stemmed?)
Happiest Song on Thriller?

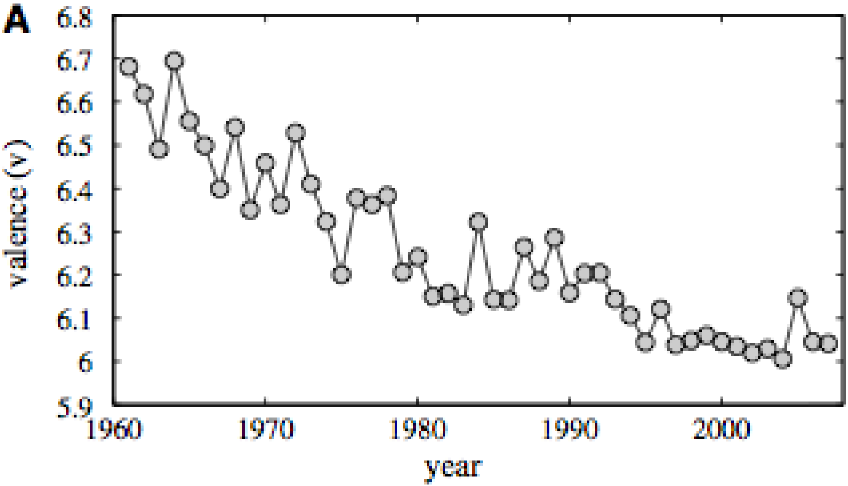


Homework Hints: One approach: write a `for` loop searching for words in dictionary (caution: is dictionary stemmed?)

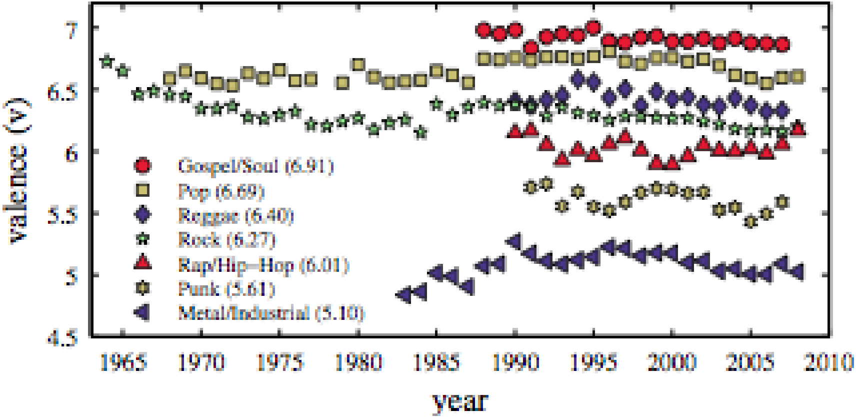
Happiest Song on Thriller?

P.Y.T. (Pretty Young Thing) (This is the right answer!)

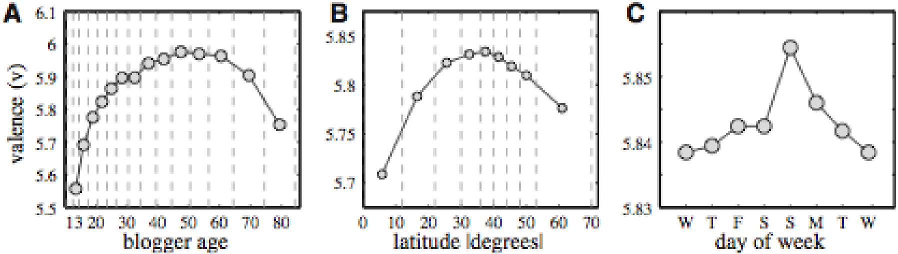
Happiness in Society



Happiness in Society



Happiness in Society



Supervised Learning

Supervised Learning

Supervised Methods:

Supervised Learning

Supervised Methods:

- Models for categorizing texts

Supervised Learning

Supervised Methods:

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

Supervised Learning

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

Supervised Learning

- How to generate valid hand coding categories

Supervised Learning

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

Supervised Learning

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

Supervised Learning

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

Supervised Learning

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

Methods generalize beyond text

Components to Supervised Learning Method

Components to Supervised Learning Method

1) Set of categories

Components to Supervised Learning Method

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

Components to Supervised Learning Method

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

Components to Supervised Learning Method

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

Components to Supervised Learning Method

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

Components to Supervised Learning Method

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

How Do We Generate Coding Rules and Categories?

How Do We Generate Coding Rules and Categories?

Challenge: coding rules/training coders to maximize coder performance

How Do We Generate Coding Rules and Categories?

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

How Do We Generate Coding Rules and Categories?

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

1) Limits of Humans:

How Do We Generate Coding Rules and Categories?

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

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

How Do We Generate Coding Rules and Categories?

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

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

How Do We Generate Coding Rules and Categories?

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

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

How Do We Generate Coding Rules and Categories?

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

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

For supervised methods to work: maximize coder agreement (without cheating!)

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 (without cheating!)

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 (without cheating!)

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 (without cheating!)

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

How Do We Generate Coding Rules?

Iterative process for generating coding rules:

How Do We Generate Coding Rules?

Iterative process for generating coding rules:

1) Write a set of coding rules

How Do We Generate Coding Rules?

Iterative process for generating coding rules:

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

How Do We Generate Coding Rules?

Iterative process for generating coding rules:

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

How Do We Generate Coding Rules?

Iterative process for generating coding rules:

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

How Do We Identify Coding Disagreement?

Many measures of inter-coder agreement

Essentially attempt to summarize a confusion matrix

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

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

Generalize across (k) coders:

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

How Do We Identify Coding Disagreements?

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

- Ambiguity
- Coder slacking

Example: 3 Coders, 8 categories.

How Do We Identify Coding Disagreements?

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

- Ambiguity
- Coder slacking

Example: 3 Coders, 8 categories.

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

How Do We Identify Coding Disagreements?

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

- Ambiguity
- Coder slacking

Example: 3 Coders, 8 categories.

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

How Do We Identify Coding Disagreements?

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

- Ambiguity
- Coder slacking

Example: 3 Coders, 8 categories.

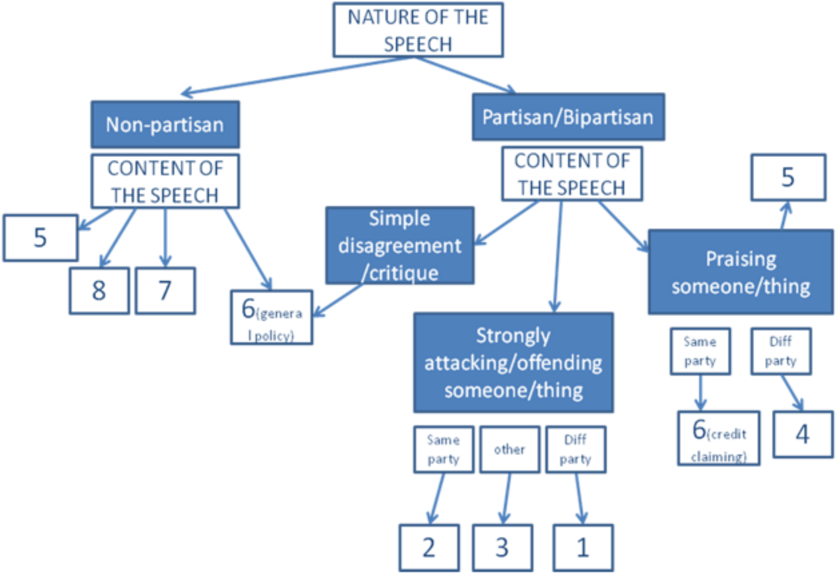
	Coder C								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	Tota
Coder B									
1	**18**	0	1	0	0	0	0	0	
3	1	0	**1**	0	0	0	0	0	
4	0	0	1	**7**	0	1	0	0	
5	0	0	0	2	**18**	3	0	0	
6	13	1	7	4	1	**26**	0	0	
7	3	0	0	0	0	8	**63**	2	
8	0	0	0	0	0	4	1	**15**	
Total	35	1	10	13	19	42	64	17	

Example Coding Document

8 part coding scheme

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

Example Coding Document



How Do We Summarize Confusion Matrix?

Lots of statistics to summarize confusion matrix:

- Most common: intercoder agreement

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

Liberal measure of agreement:

Liberal measure of agreement:

- Some agreement by chance

Liberal measure of agreement:

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

Liberal measure of agreement:

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

Liberal measure of agreement:

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

Liberal measure of agreement:

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

What to do?

Liberal measure of agreement:

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

What to do?

Suggestion: Subtract off amount expected by chance:

Liberal measure of agreement:

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

What to do?

Suggestion: Subtract off amount expected by chance:

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

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(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)

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 (From Dan Jurafsky)

Training size

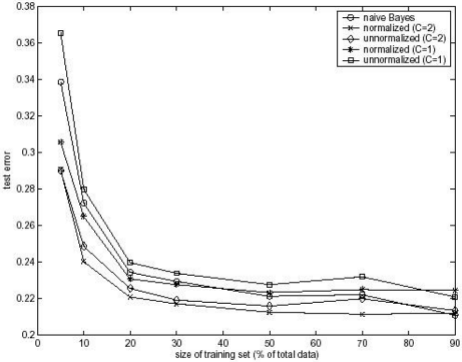


Figure 2: Test error vs training size on the newsgroups alt.atheism and talk.religion.misc

Three categories of documents

Hand labeled

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

Unlabeled

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

Label more documents than necessary to train model

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow$ {liberal, conservative}

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{liberal, conservative}\}$
We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{liberal, conservative}\}$
We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$f(\beta, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta' \boldsymbol{x}_i\right)^2$$

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{liberal, conservative}\}$
We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \\
\widehat{\boldsymbol{\beta}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \right\}
\end{aligned}
$$

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{liberal, conservative}\}$
We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}^{'} \boldsymbol{x}_i \right)^2 \\
\widehat{\boldsymbol{\beta}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}^{'} \boldsymbol{x}_i \right)^2 \right\} \\
&= \left(\boldsymbol{X}^{'} \boldsymbol{X} \right)^{-1} \boldsymbol{X}^{'} \boldsymbol{Y}
\end{aligned}
$$

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{liberal, conservative}\}$
We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}^{'} \boldsymbol{x}_i\right)^2 \\
\widehat{\boldsymbol{\beta}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}^{'} \boldsymbol{x}_i\right)^2 \right\} \\
&= \left(\boldsymbol{X}^{'} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^{'} \boldsymbol{Y}
\end{aligned}
$$

Problem:

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{liberal, conservative}\}$
We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}^{'} \boldsymbol{x}_i\right)^2 \\
\widehat{\boldsymbol{\beta}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}^{'} \boldsymbol{x}_i\right)^2 \right\} \\
&= \left(\boldsymbol{X}^{'} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^{'} \boldsymbol{Y}
\end{aligned}
$$

Problem:

- J will likely be large (perhaps $J > N$)

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{liberal, conservative}\}$
We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}^{'} \boldsymbol{x}_i\right)^2 \\
\widehat{\boldsymbol{\beta}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}^{'} \boldsymbol{x}_i\right)^2 \right\} \\
&= \left(\boldsymbol{X}^{'} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^{'} \boldsymbol{Y}
\end{aligned}
$$

Problem:

- J will likely be large (perhaps $J > N$)
- There many correlated variables

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{liberal, conservative}\}$
We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}^{'} \boldsymbol{x}_i\right)^2 \\
\widehat{\boldsymbol{\beta}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}^{'} \boldsymbol{x}_i\right)^2 \right\} \\
&= \left(\boldsymbol{X}^{'} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^{'} \boldsymbol{Y}
\end{aligned}
$$

Problem:

- J will likely be large (perhaps $J > N$)
- There many correlated variables

Predictions will be variable

Lasso Regression Objective Function/Optimization

Penalty for Model Complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \underbrace{|\beta_j|}_{\text{Penalty}}$$

Lasso Regression Objective Function/Optimization

Penalty for Model Complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \underbrace{|\beta_j|}_{\text{Penalty}}$$

- Optimization is non-linear (Absolute Value)

Lasso Regression Objective Function/Optimization

Penalty for Model Complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \underbrace{|\beta_j|}_{\text{Penalty}}$$

- Optimization is non-linear (Absolute Value)
    - Coordinate Descent

Lasso Regression Objective Function/Optimization

Penalty for Model Complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \underbrace{|\beta_j|}_{\text{Penalty}}$$

- Optimization is non-linear (Absolute Value)
  - Coordinate Descent
  - Start with Ridge

Lasso Regression Objective Function/Optimization

Penalty for Model Complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \underbrace{|\beta_j|}_{\text{Penalty}}$$

- Optimization is non-linear (Absolute Value)
  - Coordinate Descent
  - Start with Ridge
  - Sub-differential, update steps

Lasso Regression Objective Function/Optimization

Penalty for Model Complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \underbrace{|\beta_j|}_{\text{Penalty}}$$

- Optimization is non-linear (Absolute Value)
  - Coordinate Descent
  - Start with Ridge
  - Sub-differential, update steps
- Induces sparsity $\rightsquigarrow$ sets some coefficients to zero

Selecting λ

How do we determine λ? $\rightsquigarrow$ Cross validation

`To the R code!`

Selecting λ

How do we determine λ? $\rightsquigarrow$ Cross validation
Applying models gives score (probability) of document belong to class$\rightsquigarrow$ threshold to classify

`To the R code!`

Selecting λ

How do we determine λ? $\rightsquigarrow$ Cross validation

Applying models gives score (probability) of document belong to class $\rightsquigarrow$ threshold to classify

`To the R code!`

Assessing Models (Elements of Statistical Learning)

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

Comparing Training and Validation Set

Text classification and model assessment

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

Confusion matrix

Comparing Training and Validation Set

Representation of Test Statistics from 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

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

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

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

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

Comparing Training and Validation Set

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

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

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

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:

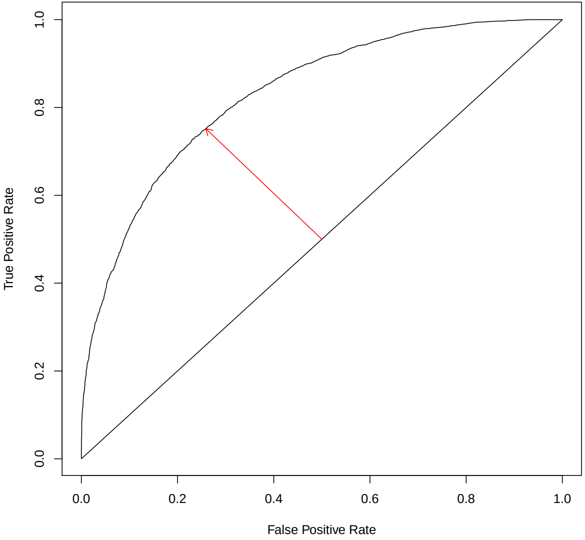
- Everything liberal: $\text{Recall}_{\text{Liberal}}$ =1 ; $\text{Recall}_{\text{Conservative}} = 0$
- Everything conservative: $\text{Recall}_{\text{Liberal}}$ =0 ; $\text{Recall}_{\text{Conservative}} = 1$

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, Westwood, and Messing (2014)

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

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

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

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

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

$$(0.1)$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

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 (Jurafsky Inspired Slide)

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 (Jurafsky Inspired Slide)

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 $p(C_k|\boldsymbol{x}_i)$.

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

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

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 (Jurafsky Inspired Slide)

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 $p(C_k|\boldsymbol{x}_i)$.

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

Naive Bayes and Optimization (Jurafsky Inspired Slide)

Naive Bayes and Optimization (Jurafsky Inspired Slide)

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

Naive Bayes and Optimization (Jurafsky Inspired Slide)

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

Naive Bayes and Optimization (Jurafsky Inspired Slide)

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

Naive Bayes and Optimization (Jurafsky Inspired Slide)

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

Two probabilities to estimate:

Naive Bayes and Optimization (Jurafsky Inspired Slide)

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

Two probabilities to estimate:

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

Naive Bayes and Optimization (Jurafsky Inspired Slide)

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

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

Naive Bayes and Optimization (Jurafsky Inspired Slide)

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

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 (Jurafsky Inspired Slide)

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

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

Naive Bayes and Optimization (Jurafsky Inspired Slide)

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

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

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

Naive Bayes and Optimization (Jurafsky Inspired Slide)

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 (Jurafsky Inspired Slide)

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 (Jurafsky Inspired Slide)

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):

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

Naive Bayes and Optimization (Jurafsky Inspired Slide)

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 (Jurafsky Inspired Slide)

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):

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

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

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

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

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

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 (Jurafsky Inspired Slide)

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 (Jurafsky Inspired Slide)

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 (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

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

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 (Jurafsky Inspired Slide)

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 (Jurafsky Inspired Slide)

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 (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

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

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

Assume the following data generating process (should look familiar)

$$
\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\
\boldsymbol{\theta} &\sim \text{Dirichlet}(\boldsymbol{\lambda}) \\
\boldsymbol{\tau}_i &\sim \text{Multinomial}(1, \boldsymbol{\pi}) \\
\boldsymbol{x}_i | \tau_{ik} = 1, \boldsymbol{\theta} &\sim \text{Multinomial}(n_i, \boldsymbol{\theta}_k)
\end{aligned}
$$

Naive Bayes and Unigram Language Models

Assume the following data generating process (should look familiar)

$$\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\
\boldsymbol{\theta} &\sim \text{Dirichlet}(\boldsymbol{\lambda}) \\
\boldsymbol{\tau}_i &\sim \text{Multinomial}(1, \boldsymbol{\pi}) \\
\boldsymbol{x}_i | \tau_{ik} = 1, \boldsymbol{\theta} &\sim \text{Multinomial}(n_i, \boldsymbol{\theta}_k)
\end{aligned}$$

If we randomly sample documents N_{train} and label them ($\boldsymbol{Y}$), then we can estimate

Naive Bayes and Unigram Language Models

Assume the following data generating process (should look familiar)

$$
\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\
\boldsymbol{\theta} &\sim \text{Dirichlet}(\boldsymbol{\lambda}) \\
\boldsymbol{\tau}_i &\sim \text{Multinomial}(1, \boldsymbol{\pi}) \\
\boldsymbol{x}_i | \tau_{ik} = 1, \boldsymbol{\theta} &\sim \text{Multinomial}(n_i, \boldsymbol{\theta}_k)
\end{aligned}
$$

If we randomly sample documents N_{train} and label them $(\boldsymbol{Y})$, then we can estimate

$$
\widehat{\pi}_k = \frac{\sum_{i=1}^{N} I(Y_i = k) + \alpha_k}{N_{\text{train}} + \sum_{k=1}^{K} \alpha_k}
$$

Naive Bayes and Unigram Language Models

Assume the following data generating process (should look familiar)

$$
\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\
\boldsymbol{\theta} &\sim \text{Dirichlet}(\boldsymbol{\lambda}) \\
\boldsymbol{\tau}_i &\sim \text{Multinomial}(1, \boldsymbol{\pi}) \\
\boldsymbol{x}_i | \tau_{ik} = 1, \boldsymbol{\theta} &\sim \text{Multinomial}(n_i, \boldsymbol{\theta}_k)
\end{aligned}
$$

If we randomly sample documents N_{train} and label them $(\boldsymbol{Y})$, then we can estimate

$$
\begin{aligned}
\widehat{\pi}_k &= \frac{\sum_{i=1}^{N} I(Y_i = k) + \alpha_k}{N_{\text{train}} + \sum_{k=1}^{K} \alpha_k} \\
\widehat{\theta}_{jk} &= \frac{\sum_{i=1}^{N} I(Y_i = k) x_{ij} + \lambda_j}{\sum_{j=1}^{J} \sum_{i=1}^{N} I(Y_i = k) x_{ij} + \sum_{j=1}^{J} \lambda_j}
\end{aligned}
$$

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,]))
```

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

Naive Bayes, LASSO, ...: focused on individual document classification.

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

Naive Bayes, LASSO, ...: focused on individual document classification.
But what if we're focused on proportions only?

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

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

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

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

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

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

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

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

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

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

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

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

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

Basic intuition:

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

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

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

Basic intuition:

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

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

Measure only presence/absence of each term [(Jx1) vector]

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

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

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

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

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

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

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

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

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

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

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$P(\boldsymbol{x}) = \text{probability of observing } \boldsymbol{x}$$

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

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$P(\boldsymbol{x}) = \text{probability of observing } \boldsymbol{x}$$
$$P(\boldsymbol{x}|C_j) = \text{Probability of observing } \boldsymbol{x} \text{ conditional on category } C_j$$

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

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

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

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

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

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

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

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

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

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

$$\underbrace{P(\boldsymbol{x})}_{2^J x 1} = \underbrace{P(\boldsymbol{x}|C)}_{2^J x K}\underbrace{P(C)}_{K x 1}$$

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

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

Algorithm Summarized

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

Assessing Model Performance

Not classifying individual documents $\rightarrow$ different standards
Mean Square Error :

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

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

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

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

Visualize: plot true and estimated proportions

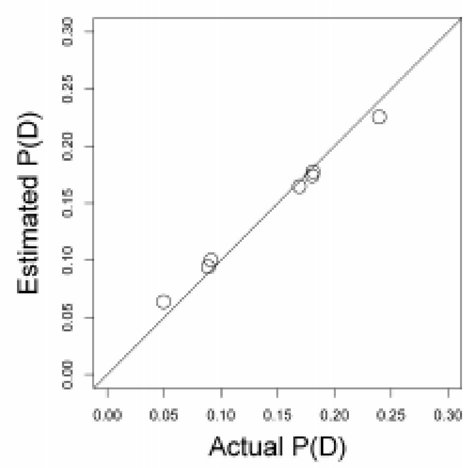

Estimated P(D)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Actual P(D)

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

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

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

Using the House Press Release Data

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

Code to Run in R

Control file:

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

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