Text as Data

Justin Grimmer

Professor
Department of Political Science
Stanford University

May 2nd, 2019

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
2) Simultaneously discover categories and measure prevalence

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

  ⇒ {Agriculture, Crime, Environment, ...}

- Campaign agendas

  ⇒ {Abortion, Campaign, Finance, Taxing, ... }

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

- Positions on legislation

  ⇒ { Support, Ambiguous, Oppose }

- Positions on Court Cases

  ⇒ { Agree with Court, Disagree with Court }

- Liberal/Conservative Blog Posts

  ⇒ { 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 an 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 of 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

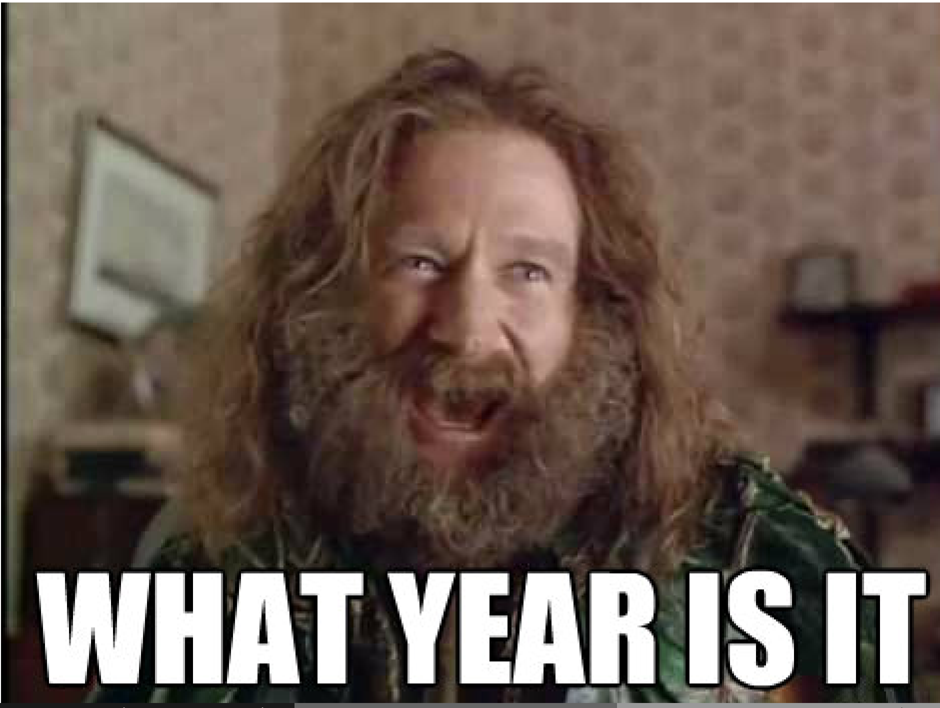
> "provides both social scientific and humanistic understandings"
>
> —Don Waisanen, Baruch College

Pre-existing word weights⇝ 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)

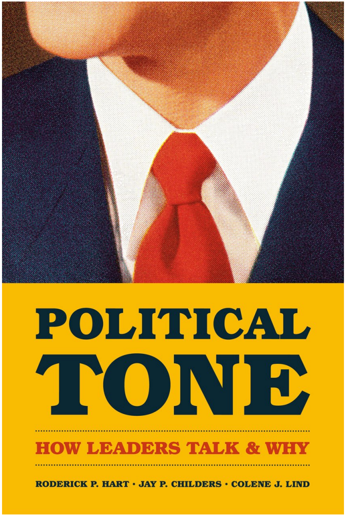
1) Proprietary $\rightsquigarrow$ wrapped in GUI
2) Basic tasks:
    a) Count words
    b) Weighted counts of words
    c) Some graphics

Dictionary Methods

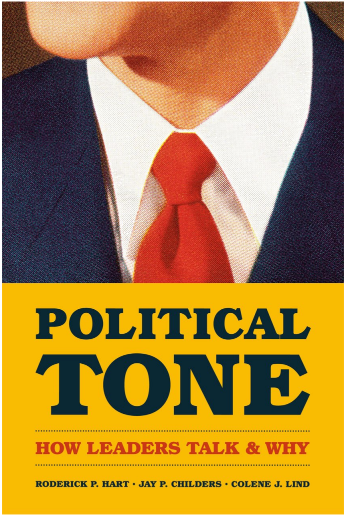
Many Dictionary Methods (like DICTION)

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

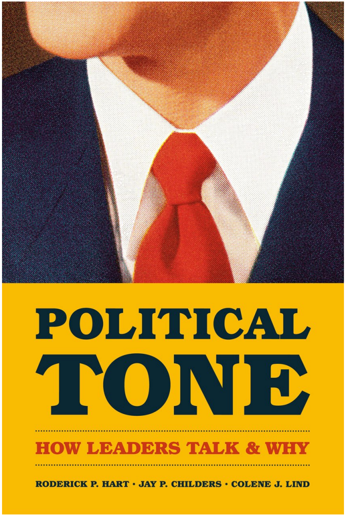

POLITICAL
TONE
HOW LEADERS TALK & WHY
RODERICK P. HART · JAY P. CHILDERS · COLENE J. LIND

DICTION



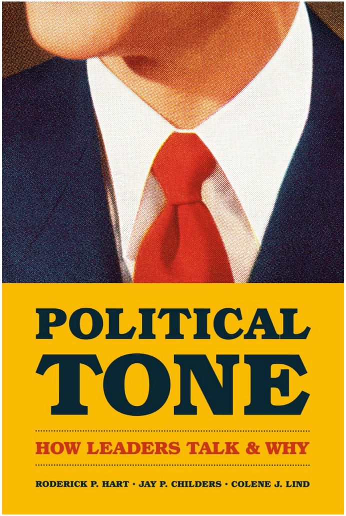
- { Certain, Uncertain }

DICTION



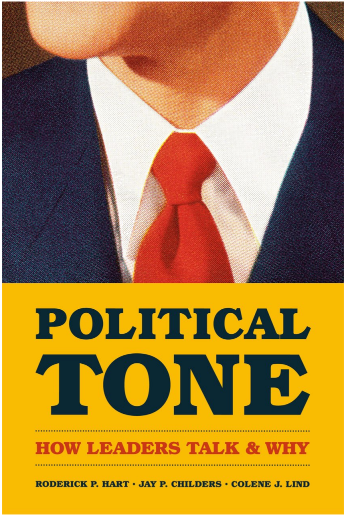
- { Certain, Uncertain }
, { Optimistic, Pessimistic }

DICTION



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

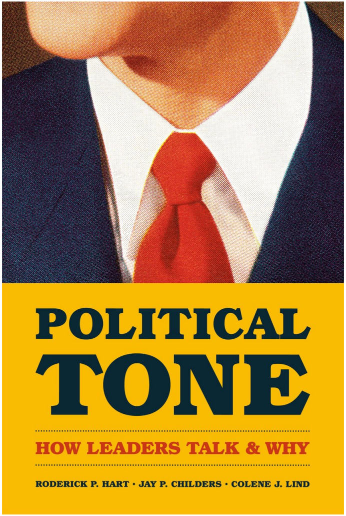
DICTION



- { 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

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

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

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

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

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

$Y_i \approx$ 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

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

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

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

$Y_i > 0 \Rightarrow$ 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

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

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

$Y_i > 0 \Rightarrow$ Positive Category

$Y_i < 0 \Rightarrow$ Negative Category

$Y_i \approx 0$ Ambiguous

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

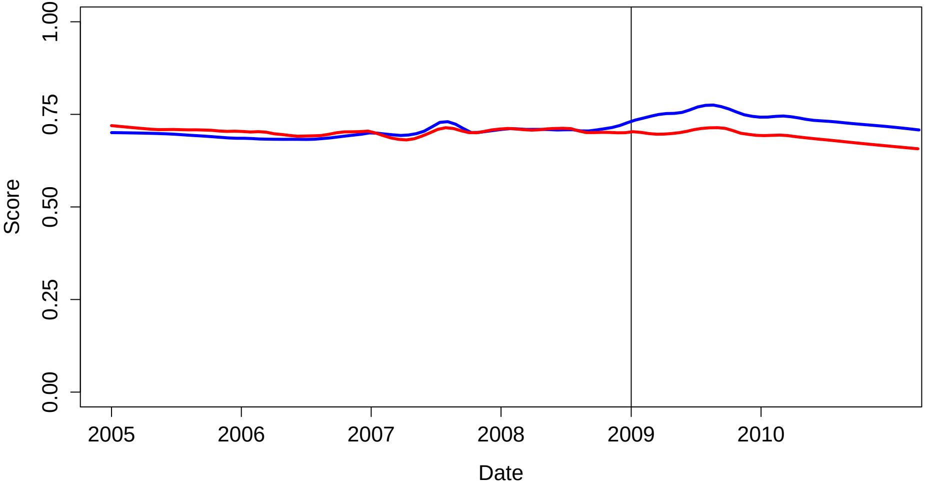
Examining Positive and Negative Statements in Press Releases

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



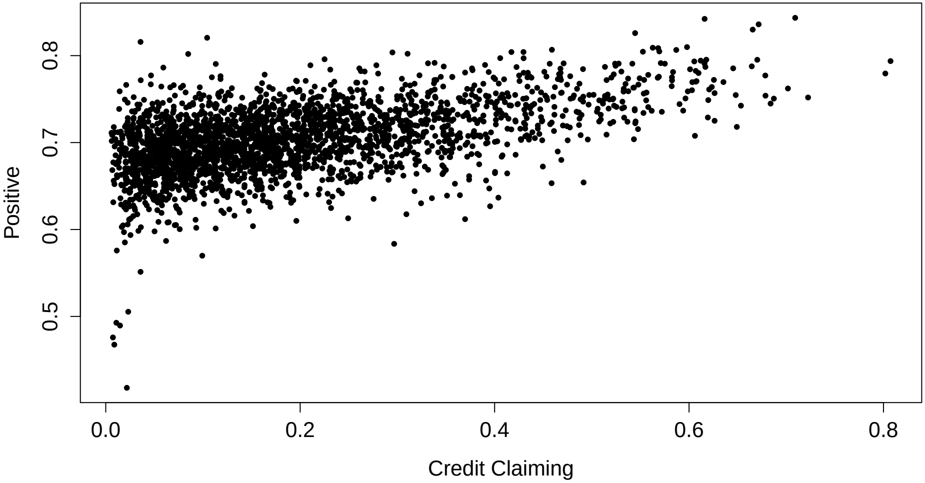
Examining Positive and Negative Statements in Press Releases

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

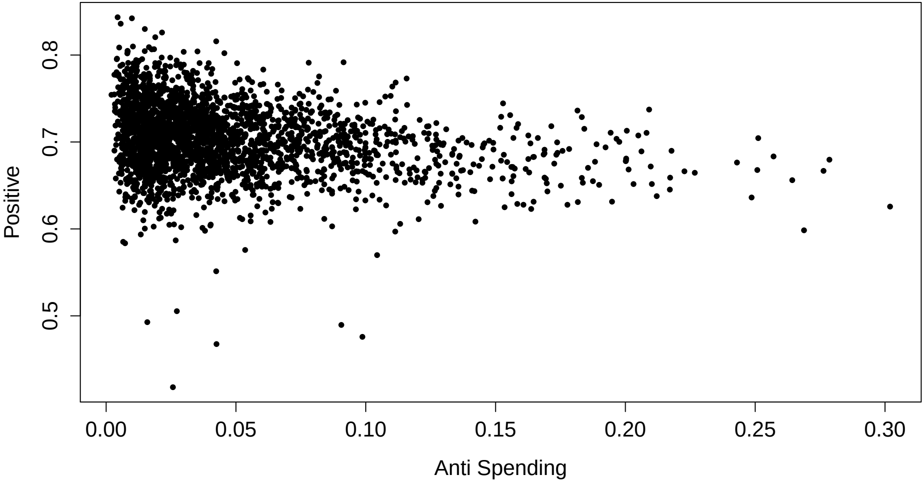
Examining Positive and Negative Statements in Press Releases

- 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

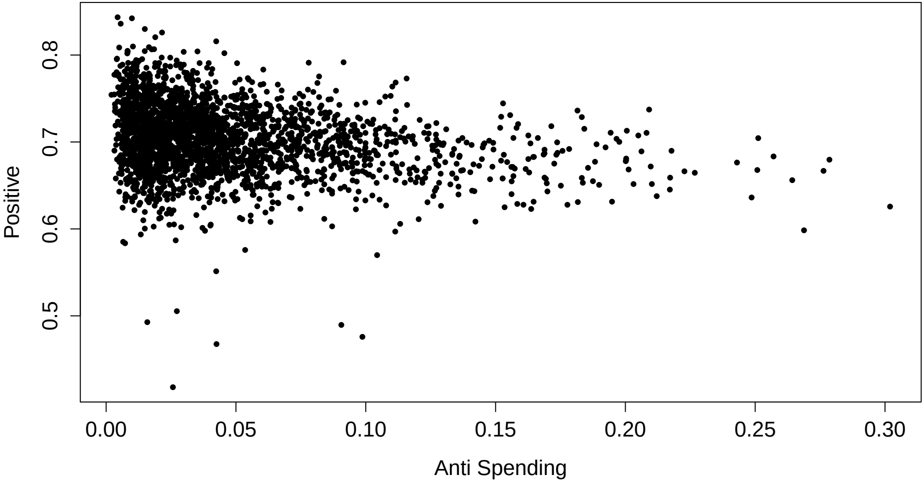
Examining Positive and Negative Statements in Press Releases



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

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

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

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

Assessing Classification

Measures of classification performance

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

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

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

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

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

Assessing Classification

Measures of classification performance

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

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

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

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

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

Under reported for dictionary classification

What about continuous measures?

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise

$\rightsquigarrow$

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)

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

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

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

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

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

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

Lower level classification

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

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

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

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

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

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

What about continuous measures?

Necessarily more complicated

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

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

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

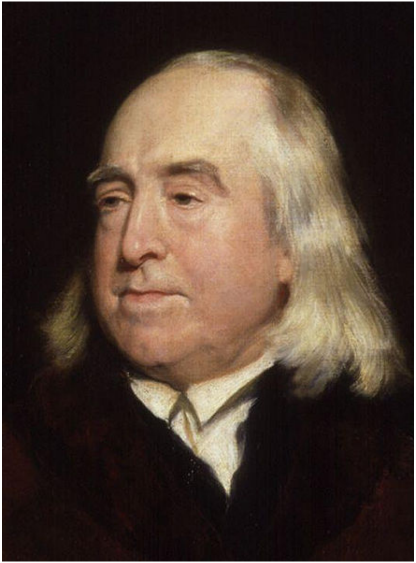
- tone matters ($)

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

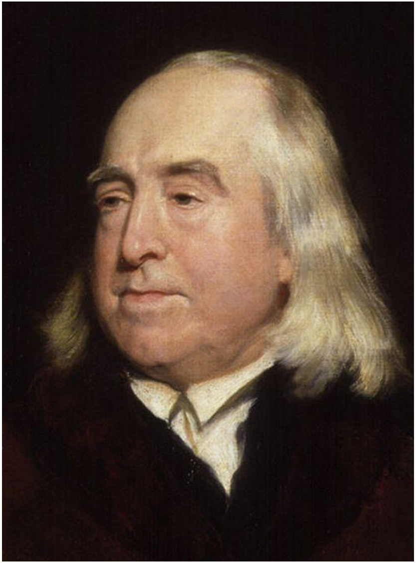
Loughran and McDonald (2011): Financial Documents are Different, polysemes

- 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, unanticipated`

Measuring Happiness

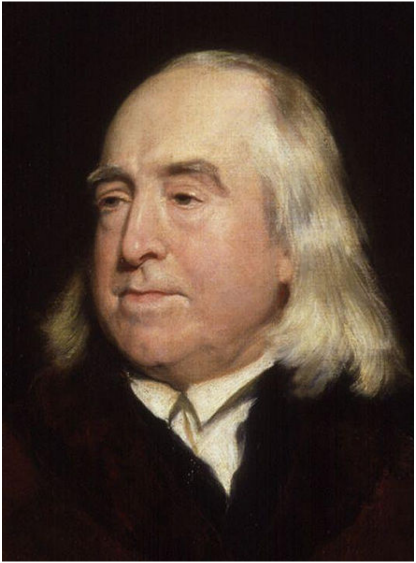


Measuring Happiness



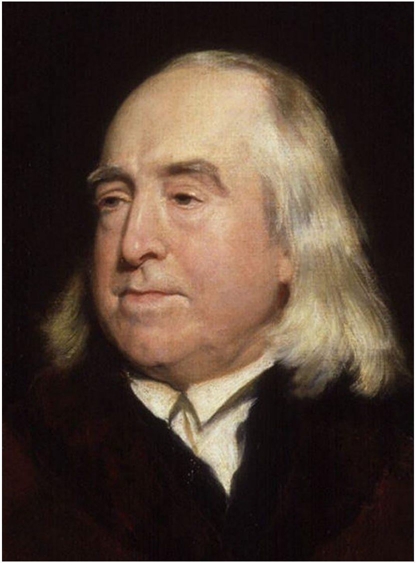
- Quantifying Happiness: How happy is society?

Measuring Happiness



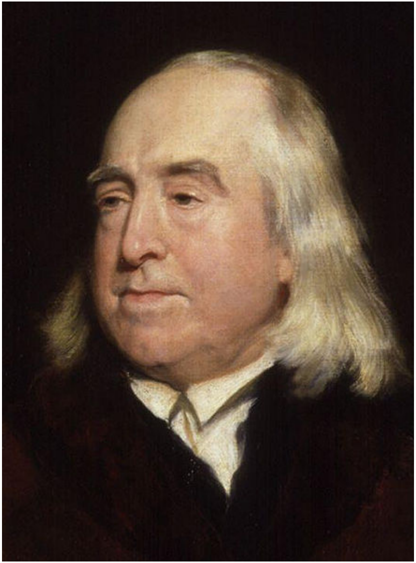
- Quantifying Happiness: How happy is society?
- How Happy is a Song?

Measuring Happiness



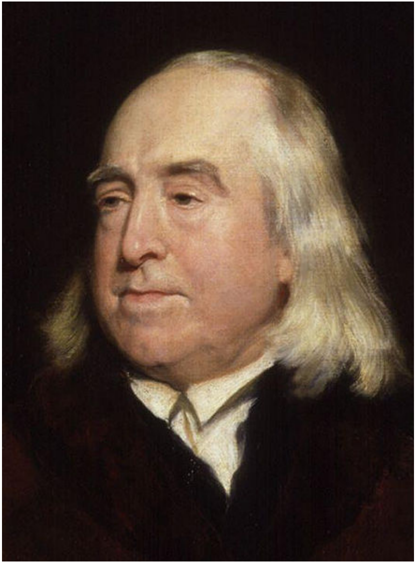
- Quantifying Happiness: How happy is society?
- How Happy is a Song?
- Blog posts?

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.
⋮

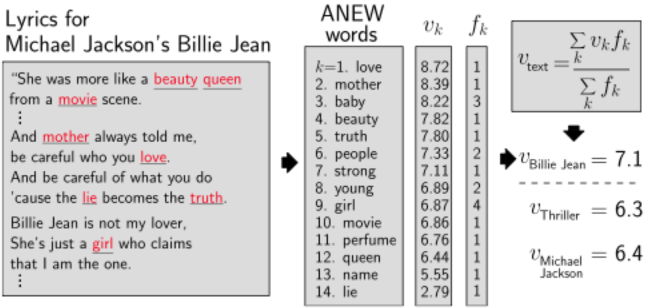
ANEW words	v_k	f_k
k=1. love	8.72	1
2. mother	8.39	1
3. baby	8.22	3
4. beauty	7.82	1
5. truth	7.80	1
6. people	7.33	2
7. strong	7.11	1
8. young	6.89	2
9. girl	6.87	4
10. movie	6.86	1
11. perfume	6.76	1
12. queen	6.44	1
13. name	5.55	1
14. lie	2.79	1

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

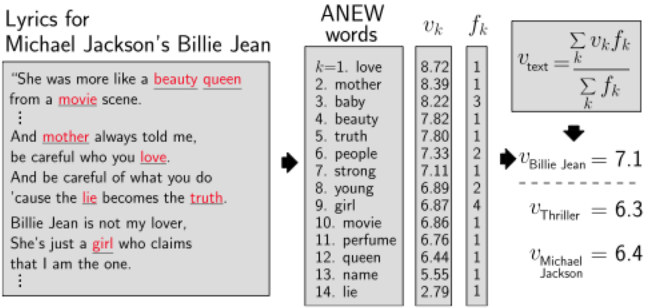
$$v_{\text{Billie Jean}} = 7.1$$

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

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

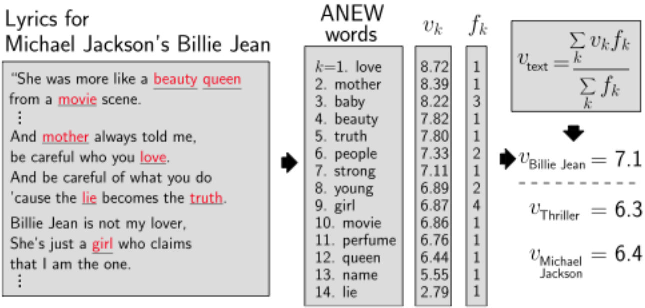


Homework Hints: One approach: write a `for` loop searching for words in dictionary (caution: is dictionary stemmed?)



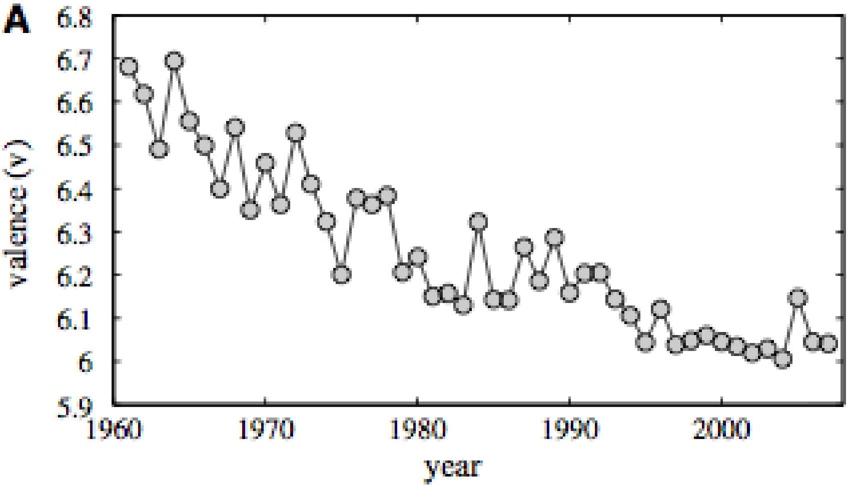
Homework Hints: One approach: write a `for` loop searching for words in dictionary (caution: is dictionary stemmed?)

Happiest Song on Thriller?

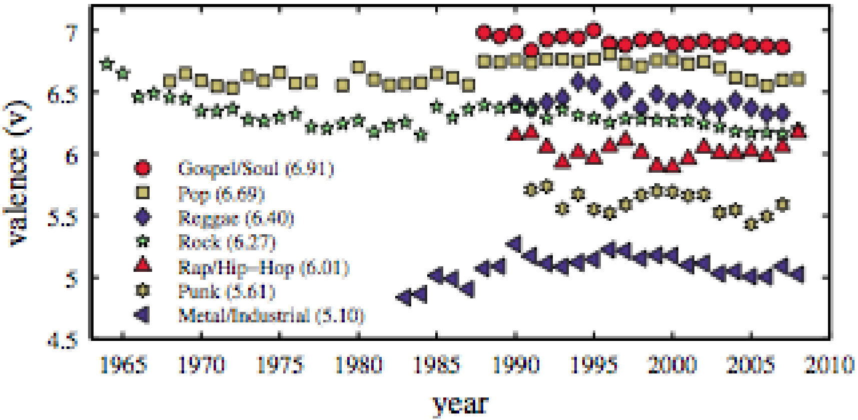


Homework Hints: One approach: write a `for` loop searching for words in dictionary (caution: is dictionary stemmed?)
Happiest Song on Thriller?
P.Y.T. (Pretty Young Thing) (This is the right answer!)

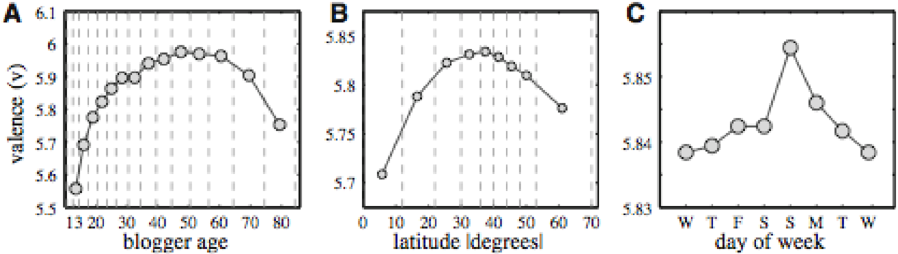
Happiness in Society



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

The Unreliability of Measures of Intercoder Reliability

The Unreliability of Measures of Intercoder Reliability

1) Hand Coding $\rightsquigarrow$ proportion in categories

The Unreliability of Measures of Intercoder Reliability

1) Hand Coding $\rightsquigarrow$ proportion in categories
2) Hand Coding (training set), machine classification $\rightsquigarrow$ proportion in categories

The Unreliability of Measures of Intercoder Reliability

1) Hand Coding $\rightsquigarrow$ proportion in categories
2) Hand Coding (training set), machine classification$\rightsquigarrow$ proportion in categories
3) Perfect training set (keywords, metadata) , machine classification$\rightsquigarrow$ proportion in categories

The Unreliability of Measures of Intercoder Reliability

1) Hand Coding $\rightsquigarrow$ proportion in categories
2) Hand Coding (training set), machine classification$\rightsquigarrow$ proportion in categories
3) Perfect training set (keywords, metadata) , machine classification$\rightsquigarrow$ proportion in categories

Usual Procedure:

The Unreliability of Measures of Intercoder Reliability

1) Hand Coding $\rightsquigarrow$ proportion in categories
2) Hand Coding (training set), machine classification$\rightsquigarrow$ proportion in categories
3) Perfect training set (keywords, metadata) , machine classification$\rightsquigarrow$ proportion in categories

Usual Procedure:

- Pay attention to percent agreement $\rightsquigarrow$ reliability

The Unreliability of Measures of Intercoder Reliability

1) Hand Coding $\rightsquigarrow$ proportion in categories
2) Hand Coding (training set), machine classification$\rightsquigarrow$ proportion in categories
3) Perfect training set (keywords, metadata) , machine classification$\rightsquigarrow$ proportion in categories

Usual Procedure:

- Pay attention to percent agreement $\rightsquigarrow$ reliability
- Set arbitrary reliability threshold$\rightsquigarrow$ ignore remaining coder disagreement

The Unreliability of Measures of Intercoder Reliability

1) Hand Coding $\rightsquigarrow$ proportion in categories
2) Hand Coding (training set), machine classification$\rightsquigarrow$ proportion in categories
3) Perfect training set (keywords, metadata) , machine classification$\rightsquigarrow$ proportion in categories

Usual Procedure:

- Pay attention to percent agreement $\rightsquigarrow$ reliability
- Set arbitrary reliability threshold$\rightsquigarrow$ ignore remaining coder disagreement
- Fit Annotation model (Dawid and Skene 1979), infer parameters

Problem:

Problem:

Coder Error $\rightsquigarrow$ Biased proportions

Problem:

- Coder Error $\rightsquigarrow$ Biased proportions
- Consequences for Business, Government, and Researchers

Problem:

- Coder Error $\rightsquigarrow$ Biased proportions
- Consequences for Business, Government, and Researchers

Solution:

Problem:

- Coder Error $\rightsquigarrow$ Biased proportions
- Consequences for Business, Government, and Researchers

Solution:

- Method and easy to use software $\rightsquigarrow$ bounds on truth

What To Do About It

Measuring reliability $\rightsquigarrow$ descriptive task

What To Do About It

Measuring reliability $\rightsquigarrow$ descriptive task
Relationship between reliability and validity $\rightsquigarrow$ inferential task

What To Do About It

Measuring reliability $\rightsquigarrow$ descriptive task

Relationship between reliability and validity $\rightsquigarrow$ inferential task

Inferential tools relating reliability and validity

What To Do About It

Measuring reliability $\rightsquigarrow$ descriptive task

Relationship between reliability and validity $\rightsquigarrow$ inferential task

Inferential tools relating reliability and validity

- Derive bounds on proportions, reliability $\leftrightarrow$ validity

What To Do About It

Measuring reliability $\rightsquigarrow$ descriptive task

Relationship between reliability and validity $\rightsquigarrow$ inferential task

Inferential tools relating reliability and validity

- Derive bounds on proportions, reliability $\leftrightarrow$ validity
  - Clear assumptions $\Rightarrow$ that bounds contain truth

What To Do About It

Measuring reliability $\rightsquigarrow$ descriptive task

Relationship between reliability and validity $\rightsquigarrow$ inferential task

Inferential tools relating reliability and validity

- Derive bounds on proportions, reliability $\leftrightarrow$ validity
  - Clear assumptions $\Rightarrow$ that bounds contain truth
  - Bounds depend on coder agreement: $\uparrow$ agreement, $\downarrow$ narrower bounds

What To Do About It

Measuring reliability $\rightsquigarrow$ descriptive task

Relationship between reliability and validity $\rightsquigarrow$ inferential task

Inferential tools relating reliability and validity

- Derive bounds on proportions, reliability $\leftrightarrow$ validity
  - Clear assumptions $\Rightarrow$ that bounds contain truth
  - Bounds depend on coder agreement: $\uparrow$ agreement, $\downarrow$ narrower bounds
- Extensions for alternative settings and inferences

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1, 2, 3\}$$

$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$

$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$$

$$\text{truth} = \bar{\boldsymbol{\pi}}$$

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1, 2, 3\}$$

$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$

$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3) = (0.7,\ 0.25,\ 0.05)$$

$$\text{truth} = \bar{\boldsymbol{\pi}}$$

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1, 2, 3\}$$

$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$

$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$$

Coders:

$$\text{truth} = \bar{\boldsymbol{\pi}}$$

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1, 2, 3\}$$

$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$

$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$$

Coders:

$$y_d^1 \in \{1, 2, 3\}$$

truth $=$ $\bar{\boldsymbol{\pi}}$

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1, 2, 3\}$$

$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$

$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$$

Coders:

$$y_d^1 \in \{1, 2, 3\} \text{ , } y_d^2 \in \{1, 2, 3\}$$

truth $=$ $\bar{\boldsymbol{\pi}}$

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1, 2, 3\}$$

$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$

$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$$

Coders:

$$y_d^1 \in \{1, 2, 3\} \ , \ y_d^2 \in \{1, 2, 3\}$$

$$\bar{y}_k^1 = \text{mean}_d I[(y_d^1 = k)]$$

$$\bar{y}_k^2 = \text{mean}_d I[(y_d^2 = k)]$$

truth $= \bar{\boldsymbol{\pi}}$

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1, 2, 3\}$$

$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$

$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$$

Coders:

$$y_d^1 \in \{1, 2, 3\}\ ,\ y_d^2 \in \{1, 2, 3\}$$

$$\bar{y}_k^1 = \text{mean}_d I[(y_d^1 = k)]$$

$$\bar{y}_k^2 = \text{mean}_d I[(y_d^2 = k)]$$

$$\bar{y}_k = \text{mean}_c[\bar{y}_k^c]$$

truth $= \bar{\boldsymbol{\pi}}$

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1, 2, 3\}$$

$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$

$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$$

Coders:

$$y_d^1 \in \{1, 2, 3\} \ , \ y_d^2 \in \{1, 2, 3\}$$

$$\bar{y}_k^1 = \text{mean}_d I[(y_d^1 = k)]$$

$$\bar{y}_k^2 = \text{mean}_d I[(y_d^2 = k)]$$

$$\bar{y}_k = \text{mean}_c[\bar{y}_k^c]$$

$$\bar{\boldsymbol{y}} = (\bar{y}_1, \bar{y}_2, \bar{y}_3)$$

truth $= \bar{\boldsymbol{\pi}}$

naïve estimate $= \bar{\boldsymbol{y}}$

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1, 2, 3\}$$

$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$

$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$$

Coders:

$$y_d^1 \in \{1, 2, 3\} \ , \ y_d^2 \in \{1, 2, 3\}$$

$$\bar{y}_k^1 = \text{mean}_d I[(y_d^1 = k)]$$

$$\bar{y}_k^2 = \text{mean}_d I[(y_d^2 = k)]$$

$$\bar{y}_k = \text{mean}_c[\bar{y}_k^c]$$

$$\bar{\boldsymbol{y}} = (\bar{y}_1, \bar{y}_2, \bar{y}_3)$$

truth $= \bar{\boldsymbol{\pi}}$

naïve estimate $= \bar{\boldsymbol{y}}$

Agreement and Reliability

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1, 2, 3\}$$

$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$

$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$$

Coders:

$$y_d^1 \in \{1, 2, 3\} \text{ , } y_d^2 \in \{1, 2, 3\}$$

$$\bar{y}_k^1 = \text{mean}_d I[(y_d^1 = k)]$$

$$\bar{y}_k^2 = \text{mean}_d I[(y_d^2 = k)]$$

$$\bar{y}_k = \text{mean}_c[\bar{y}_k^c]$$

$$\bar{\boldsymbol{y}} = (\bar{y}_1, \bar{y}_2, \bar{y}_3)$$

$$\text{truth} = \bar{\boldsymbol{\pi}}$$

$$\text{naïve estimate} = \bar{\boldsymbol{y}}$$

Agreement and Reliability

$$m_{jk}^{12} = \text{mean}_d[I(y_d^1 = j, y_d^2 = k)]$$

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1, 2, 3\}$$

$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$

$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$$

Coders:

$$y_d^1 \in \{1, 2, 3\} \;,\; y_d^2 \in \{1, 2, 3\}$$

$$\bar{y}_k^1 = \text{mean}_d I[(y_d^1 = k)]$$

$$\bar{y}_k^2 = \text{mean}_d I[(y_d^2 = k)]$$

$$\bar{y}_k = \text{mean}_c[\bar{y}_k^c]$$

$$\bar{\boldsymbol{y}} = (\bar{y}_1, \bar{y}_2, \bar{y}_3)$$

truth $= \bar{\boldsymbol{\pi}}$

naïve estimate $= \bar{\boldsymbol{y}}$

reliability $= a^{12}$

Agreement and Reliability

$$m_{jk}^{12} = \text{mean}_d[I(y_d^1 = j, y_d^2 = k)]$$

$$a^{12} = \sum_{k=1}^{3} m_{kk}^{12}$$

Motivating Example and Notation

Suppose 2 coders classify D documents into 3 categories

Truth

$$\pi_d \in \{1,2,3\}$$
$$\bar{\pi}_k = \text{mean}_d[I(\pi_d = k)],$$
$$\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$$

Coders:

$$y_d^1 \in \{1,2,3\} \text{ , } y_d^2 \in \{1,2,3\}$$
$$\bar{y}_k^1 = \text{mean}_d I[(y_d^1 = k)]$$
$$\bar{y}_k^2 = \text{mean}_d I[(y_d^2 = k)]$$
$$\bar{y}_k = \text{mean}_c[\bar{y}_k^c]$$
$$\bar{\boldsymbol{y}} = (\bar{y}_1, \bar{y}_2, \bar{y}_3)$$

Agreement and Reliability

$$m_{jk}^{12} = \text{mean}_d[I(y_d^1 = j, y_d^2 = k)]$$
$$a^{12} = \sum_{k=1}^{3} m_{kk}^{12} \;\; = 0.7$$

$$\text{truth} = \bar{\boldsymbol{\pi}}$$
$$\text{naïve estimate} = \bar{\boldsymbol{y}}$$
$$\text{reliability} = a^{12}$$

The Link Between Truth and Coders' Decisions

The Link Between Truth and Coders' Decisions

Coding task $\rightsquigarrow$ map from truth to codes

The Link Between Truth and Coders' Decisions

Coding task $\rightsquigarrow$ map from truth to codes

$$\epsilon^1_{jk} \quad = \quad \text{Proportion coder 1 classifies a document in } j \text{ when truth is } k$$

The Link Between Truth and Coders' Decisions

Coding task $\rightsquigarrow$ map from truth to codes

$$
\begin{aligned}
\epsilon^{1}_{jk} &= \text{Proportion coder 1 classifies a document in } j \text{ when truth is } k \\
\epsilon^{1}_{kk} &= \text{Proportion coder 1 classifies a document in } k \text{ when truth is } k
\end{aligned}
$$

The Link Between Truth and Coders' Decisions

Coding task $\rightsquigarrow$ map from truth to codes

$$
\begin{aligned}
\epsilon^1_{jk} &= \text{Proportion coder 1 classifies a document in } j \text{ when truth is } k \\
\epsilon^1_{kk} &= \text{validity}
\end{aligned}
$$

Mapping from Truth to Coders' Decisions

The Link Between Truth and Coders' Decisions

Coding task $\rightsquigarrow$ map from truth to codes

$$\begin{aligned} \epsilon^1_{jk} &= \text{Proportion coder 1 classifies a document in } j \text{ when truth is } k \\ \epsilon^1_{kk} &= \text{validity} \end{aligned}$$

Mapping from Truth to Coders' Decisions

$$\bar{y}^2_1 = \epsilon^2_{11}\bar{\pi}_1 + \epsilon^2_{12}\bar{\pi}_2 + \epsilon^2_{13}\bar{\pi}_3$$

The Link Between Truth and Coders' Decisions

Coding task $\rightsquigarrow$ map from truth to codes

$$\begin{aligned} \epsilon_{jk}^{1} &= \text{Proportion coder 1 classifies a document in } j \text{ when truth is } k \\ \epsilon_{kk}^{1} &= \text{validity} \end{aligned}$$

Mapping from Truth to Coders' Decisions

$$\bar{y}_1^2 = \epsilon_{11}^2 \bar{\pi}_1 + \epsilon_{12}^2 \bar{\pi}_2 + \epsilon_{13}^2 \bar{\pi}_3$$

Proportion coder 2 places in category 1: mixture across categories

The Link Between Truth and Coders' Decisions

Coding task $\rightsquigarrow$ map from truth to codes

$$\begin{aligned} \epsilon_{jk}^{1} &= \text{Proportion coder 1 classifies a document in } j \text{ when truth is } k \\ \epsilon_{kk}^{1} &= \text{validity} \end{aligned}$$

Mapping from Truth to Coders' Decisions

$$\bar{y}_1^2 = \epsilon_{11}^2\bar{\pi}_1 + \epsilon_{12}^2\bar{\pi}_2 + \epsilon_{13}^2\bar{\pi}_3$$

Proportion coder 2 places in category 1: mixture across categories
For Example:

The Link Between Truth and Coders' Decisions

Coding task $\rightsquigarrow$ map from truth to codes

$$\begin{aligned} \epsilon^1_{jk} &= \text{Proportion coder 1 classifies a document in } j \text{ when truth is } k \\ \epsilon^1_{kk} &= \text{validity} \end{aligned}$$

Mapping from Truth to Coders' Decisions

$$\bar{y}^2_1 = \epsilon^2_{11}\bar{\pi}_1 + \epsilon^2_{12}\bar{\pi}_2 + \epsilon^2_{13}\bar{\pi}_3$$

Proportion coder 2 places in category 1: mixture across categories

For Example:

$\epsilon^2_{11} = 0.8$, $\epsilon^2_{12} = 0.14$,$\epsilon^2_{13} = 0.17$ and $\bar{\boldsymbol{\pi}} = (0.7, 0.25, 0.05)$ then

The Link Between Truth and Coders' Decisions

Coding task $\rightsquigarrow$ map from truth to codes

$$
\begin{aligned}
\epsilon^1_{jk} &= \text{Proportion coder 1 classifies a document in } j \text{ when truth is } k \\
\epsilon^1_{kk} &= \text{validity}
\end{aligned}
$$

Mapping from Truth to Coders' Decisions

$$\bar{y}^2_1 = \epsilon^2_{11}\bar{\pi}_1 + \epsilon^2_{12}\bar{\pi}_2 + \epsilon^2_{13}\bar{\pi}_3$$

Proportion coder 2 places in category 1: mixture across categories

For Example:

$\epsilon^2_{11} = 0.8$, $\epsilon^2_{12} = 0.14$,$\epsilon^2_{13} = 0.17$ and $\bar{\boldsymbol{\pi}} = (0.7, 0.25, 0.05)$ then

$$\bar{y}^2_1 = 0.8 \times 0.7 + 0.14 \times 0.25 + 0.17 \times 0.05 = 0.60$$

The Link Between Truth and Coders' Decisions

Coding task $\rightsquigarrow$ map from truth to codes

$$\begin{aligned} \epsilon_{jk}^{1} &= \text{Proportion coder 1 classifies a document in } j \text{ when truth is } k \\ \epsilon_{kk}^{1} &= \text{validity} \end{aligned}$$

Mapping from Truth to Coders' Decisions

$$\bar{y}_1^2 = \epsilon_{11}^2 \bar{\pi}_1 + \epsilon_{12}^2 \bar{\pi}_2 + \epsilon_{13}^2 \bar{\pi}_3$$

Proportion coder 2 places in category 1: mixture across categories

For Example:

$\epsilon_{11}^2 = 0.8$, $\epsilon_{12}^2 = 0.14$,$\epsilon_{13}^2 = 0.17$ and $\bar{\boldsymbol{\pi}} = (0.7, 0.25, 0.05)$ then

$$\bar{y}_1^2 = 0.8 \times 0.7 + 0.14 \times 0.25 + 0.17 \times 0.05 = 0.60$$

The Link Between Truth and Coders' Decisions

Define the evaluation matrix $\boldsymbol{E}^1$:

The Link Between Truth and Coders' Decisions

Define the evaluation matrix $\boldsymbol{E}^1$:

$$\boldsymbol{E}^1 = \begin{pmatrix} \epsilon_{11}^1 & \epsilon_{12}^1 & \epsilon_{13}^1 \\ \epsilon_{21}^1 & \epsilon_{22}^1 & \epsilon_{23}^1 \\ \epsilon_{31}^1 & \epsilon_{32}^1 & \epsilon_{33}^1 \end{pmatrix}$$

The Link Between Truth and Coders' Decisions

Define the evaluation matrix $\boldsymbol{E}^1$:

$$\boldsymbol{E}^1 = \begin{pmatrix} \epsilon^1_{11} & \epsilon^1_{12} & \epsilon^1_{13} \\ \epsilon^1_{21} & \epsilon^1_{22} & \epsilon^1_{23} \\ \epsilon^1_{31} & \epsilon^1_{32} & \epsilon^1_{33} \end{pmatrix}$$

$$\boldsymbol{E}^1 = \begin{pmatrix} 0.9 & 0.07 & 0.02 \\ 0.08 & 0.9 & 0.08 \\ 0.02 & 0.03 & 0.9 \end{pmatrix}$$

$$\boldsymbol{E}^2 = \begin{pmatrix} 0.8 & 0.14 & 0.17 \\ 0.01 & 0.80 & 0.03 \\ 0.19 & 0.06 & 0.8 \end{pmatrix}$$

The Link Between Truth and Coders' Decisions

Define the evaluation matrix $\boldsymbol{E}^1$:

$$\boldsymbol{E}^1 = \begin{pmatrix} \epsilon_{11}^1 & \epsilon_{12}^1 & \epsilon_{13}^1 \\ \epsilon_{21}^1 & \epsilon_{22}^1 & \epsilon_{23}^1 \\ \epsilon_{31}^1 & \epsilon_{32}^1 & \epsilon_{33}^1 \end{pmatrix}$$

The Link Between Truth and Coders' Decisions

Define the evaluation matrix $\boldsymbol{E}^1$:

$$\boldsymbol{E}^1 = \begin{pmatrix} \epsilon_{11}^1 & \epsilon_{12}^1 & \epsilon_{13}^1 \\ \epsilon_{21}^1 & \epsilon_{22}^1 & \epsilon_{23}^1 \\ \epsilon_{31}^1 & \epsilon_{32}^1 & \epsilon_{33}^1 \end{pmatrix}$$

Then,

The Link Between Truth and Coders' Decisions

Define the evaluation matrix $\boldsymbol{E}^1$:

$$\boldsymbol{E}^1 = \begin{pmatrix} \epsilon_{11}^1 & \epsilon_{12}^1 & \epsilon_{13}^1 \\ \epsilon_{21}^1 & \epsilon_{22}^1 & \epsilon_{23}^1 \\ \epsilon_{31}^1 & \epsilon_{32}^1 & \epsilon_{33}^1 \end{pmatrix}$$

Then,

$$\begin{aligned} \bar{\boldsymbol{y}}^1 &= \boldsymbol{E}^1 \bar{\boldsymbol{\pi}} \\ \bar{\boldsymbol{y}}^2 &= \boldsymbol{E}^2 \bar{\boldsymbol{\pi}} \end{aligned}$$

The Link Between Truth and Coders' Decisions

Define the evaluation matrix $\boldsymbol{E}^1$:

$$\boldsymbol{E}^1 = \begin{pmatrix} \epsilon_{11}^1 & \epsilon_{12}^1 & \epsilon_{13}^1 \\ \epsilon_{21}^1 & \epsilon_{22}^1 & \epsilon_{23}^1 \\ \epsilon_{31}^1 & \epsilon_{32}^1 & \epsilon_{33}^1 \end{pmatrix}$$

Then,

$$\begin{aligned} \bar{\boldsymbol{y}}^1 &= \boldsymbol{E}^1 \bar{\boldsymbol{\pi}} \\ \bar{\boldsymbol{y}}^2 &= \boldsymbol{E}^2 \bar{\boldsymbol{\pi}} \end{aligned}$$

$$\begin{aligned} \bar{\boldsymbol{y}}^1 &= (0.65, 0.28, 0.07) \\ \bar{\boldsymbol{y}}^2 &= (0.6, 0.21, 0.19) \end{aligned}$$

The Link Between Truth and Coders' Decisions

Define the evaluation matrix $\boldsymbol{E}^1$:

$$\boldsymbol{E}^1 = \begin{pmatrix} \epsilon_{11}^1 & \epsilon_{12}^1 & \epsilon_{13}^1 \\ \epsilon_{21}^1 & \epsilon_{22}^1 & \epsilon_{23}^1 \\ \epsilon_{31}^1 & \epsilon_{32}^1 & \epsilon_{33}^1 \end{pmatrix}$$

Then,

$$\begin{aligned} \bar{\boldsymbol{y}}^1 &= \boldsymbol{E}^1 \bar{\boldsymbol{\pi}} \\ \bar{\boldsymbol{y}}^2 &= \boldsymbol{E}^2 \bar{\boldsymbol{\pi}} \end{aligned}$$

If $\boldsymbol{E}^1$ and $\boldsymbol{E}^2$ are known, then

The Link Between Truth and Coders' Decisions

Define the evaluation matrix $\boldsymbol{E}^1$:

$$\boldsymbol{E}^1 = \begin{pmatrix} \epsilon^1_{11} & \epsilon^1_{12} & \epsilon^1_{13} \\ \epsilon^1_{21} & \epsilon^1_{22} & \epsilon^1_{23} \\ \epsilon^1_{31} & \epsilon^1_{32} & \epsilon^1_{33} \end{pmatrix}$$

Then,

$$\begin{aligned} \bar{\boldsymbol{y}}^1 &= \boldsymbol{E}^1 \bar{\boldsymbol{\pi}} \\ \bar{\boldsymbol{y}}^2 &= \boldsymbol{E}^2 \bar{\boldsymbol{\pi}} \end{aligned}$$

If $\boldsymbol{E}^1$ and $\boldsymbol{E}^2$ are known, then

$$\begin{aligned} \left(\boldsymbol{E}^1\right)^{-1} \bar{\boldsymbol{y}}^1 &= \bar{\boldsymbol{\pi}} \\ \left(\boldsymbol{E}^2\right)^{-1} \bar{\boldsymbol{y}}^2 &= \bar{\boldsymbol{\pi}} \end{aligned}$$

The Link Between Truth and Coders' Decisions

Define the evaluation matrix $\boldsymbol{E}^1$:

$$\boldsymbol{E}^1 = \begin{pmatrix} \epsilon_{11}^1 & \epsilon_{12}^1 & \epsilon_{13}^1 \\ \epsilon_{21}^1 & \epsilon_{22}^1 & \epsilon_{23}^1 \\ \epsilon_{31}^1 & \epsilon_{32}^1 & \epsilon_{33}^1 \end{pmatrix}$$

Then,

$$\begin{aligned} \bar{\boldsymbol{y}}^1 &= \boldsymbol{E}^1 \bar{\boldsymbol{\pi}} \\ \bar{\boldsymbol{y}}^2 &= \boldsymbol{E}^2 \bar{\boldsymbol{\pi}} \end{aligned}$$

If $\boldsymbol{E}^1$ and $\boldsymbol{E}^2$ are known, then

$$\begin{aligned} \left(\boldsymbol{E}^1\right)^{-1} \bar{\boldsymbol{y}}^1 &= \bar{\boldsymbol{\pi}} \\ \left(\boldsymbol{E}^2\right)^{-1} \bar{\boldsymbol{y}}^2 &= \bar{\boldsymbol{\pi}} \end{aligned}$$

Problem: We don't (and can't) know evaluation matrices

Agreement, Assumptions, Structure $\rightsquigarrow$ Set of Matrices

The Link Between Truth and Reliability

Goal: use coders' reliability to infer validity

The Link Between Truth and Reliability

Goal: use coders' reliability to infer validity

Define:

The Link Between Truth and Reliability

Goal: use coders' reliability to infer validity

Define:

$$\epsilon^1 = \epsilon^1_{11}\bar{\pi}_1 + \epsilon^1_{22}\bar{\pi}_2 + \epsilon^1_{33}\bar{\pi}_3$$

The Link Between Truth and Reliability

Goal: use coders' reliability to infer validity

Define:

$$\epsilon^1 = \epsilon^1_{11}\bar{\pi}_1 + \epsilon^1_{22}\bar{\pi}_2 + \epsilon^1_{33}\bar{\pi}_3$$

$\epsilon^1 \rightsquigarrow$ average validity rate

The Link Between Truth and Reliability

Goal: use coders' reliability to infer validity

Define:

$$\epsilon^1 \quad = \quad \epsilon^1_{11}\bar{\pi}_1 + \epsilon^1_{22}\bar{\pi}_2 + \epsilon^1_{33}\bar{\pi}_3$$

$\epsilon^1 \rightsquigarrow$ average validity rate

Proposition

Suppose coder 1 and coder 2 have agreement rate a^{12}.
Maximum Average Validity

$$\dot{\epsilon}^{12} \quad = \quad \frac{1 + a^{12}}{2}$$

The Link Between Truth and Reliability

Goal: use coders' reliability to infer validity

Define:

$$\epsilon^1 = \epsilon^1_{11}\bar{\pi}_1 + \epsilon^1_{22}\bar{\pi}_2 + \epsilon^1_{33}\bar{\pi}_3$$

$\epsilon^1 \rightsquigarrow$ average validity rate

Proposition

Suppose coder 1 and coder 2 have agreement rate a^{12}.
Maximum Average Validity

$$\dot{\epsilon}^{12} = \frac{1 + a^{12}}{2}$$

Equivalently $\rightsquigarrow$ maximum average validity implies:

The Link Between Truth and Reliability

Goal: use coders' reliability to infer validity

Define:

$$\epsilon^{1} = \epsilon^{1}_{11}\bar{\pi}_1 + \epsilon^{1}_{22}\bar{\pi}_2 + \epsilon^{1}_{33}\bar{\pi}_3$$

$\epsilon^{1} \rightsquigarrow$ average validity rate

Proposition

Suppose coder 1 and coder 2 have agreement rate a^{12}.
Maximum Average Validity

$$\dot{\epsilon}^{12} = \frac{1 + a^{12}}{2}$$

Equivalently $\rightsquigarrow$ maximum average validity implies:

- Coders agree: correct

The Link Between Truth and Reliability

Goal: use coders' reliability to infer validity

Define:

$$\epsilon^1 = \epsilon^1_{11}\bar{\pi}_1 + \epsilon^1_{22}\bar{\pi}_2 + \epsilon^1_{33}\bar{\pi}_3$$

$\epsilon^1 \rightsquigarrow$ average validity rate

Proposition

Suppose coder 1 and coder 2 have agreement rate a^{12}. Maximum Average Validity

$$\dot{\epsilon}^{12} = \frac{1 + a^{12}}{2}$$

Equivalently $\rightsquigarrow$ maximum average validity implies:

- Coders agree: correct
- Coders disagree: at least one coder is correct

Deriving Bounds on Proportions

Assumption

Wisdom of the Coders *Coder 1 and 2 have maximum validity given their agreement rate* a^{12}

$$(0.1)$$

$$(0.2)$$

Deriving Bounds on Proportions

Assumption

Wisdom of the Coders *Coder 1 and 2 have maximum validity given their agreement rate* a^{12}

Assumption

Constant Validity Assumption *Coder c's validity is constant across categories.* $\epsilon^c = \epsilon^c_{kk}$

$$ \tag{0.1} $$

$$ \tag{0.2} $$

Deriving Bounds on Proportions

Assumption

Wisdom of the Coders *Coder 1 and 2 have maximum validity given their agreement rate* a^{12}

Assumption

Constant Validity Assumption *Coder c's validity is constant across categories.* $\epsilon^c = \epsilon^c_{kk}$

Other structure:

$$(0.1)$$

$$(0.2)$$

Deriving Bounds on Proportions

Assumption

Wisdom of the Coders *Coder 1 and 2 have maximum validity given their agreement rate* a^{12}

Assumption

Constant Validity Assumption *Coder c's validity is constant across categories.* $\epsilon^c = \epsilon^c_{kk}$

Other structure:

$$\left(\boldsymbol{E}^1\right)^{-1} \bar{\boldsymbol{y}}^1 = \bar{\boldsymbol{\pi}}$$

(0.1)

(0.2)

Deriving Bounds on Proportions

Assumption

Wisdom of the Coders *Coder 1 and 2 have maximum validity given their agreement rate* a^{12}

Assumption

Constant Validity Assumption *Coder c's validity is constant across categories.* $\epsilon^{c} = \epsilon^{c}_{kk}$

Other structure:

$$\left(\boldsymbol{E}^1\right)^{-1} \bar{\boldsymbol{y}}^1 = \bar{\boldsymbol{\pi}} \quad (0.1)$$

$$\left(\boldsymbol{E}^2\right)^{-1} \bar{\boldsymbol{y}}^2 = \bar{\boldsymbol{\pi}} \quad (0.2)$$

Deriving Bounds on Proportions

Assumption

Wisdom of the Coders *Coder 1 and 2 have maximum validity given their agreement rate* a^{12}

Assumption

Constant Validity Assumption *Coder c's validity is constant across categories.* $\epsilon^{c} = \epsilon^{c}_{kk}$

Other structure:

$$
\begin{aligned}
\left(\boldsymbol{E}^1\right)^{-1}\bar{\boldsymbol{y}}^1 &= \bar{\boldsymbol{\pi}} \\
\left(\boldsymbol{E}^2\right)^{-1}\bar{\boldsymbol{y}}^2 &= \bar{\boldsymbol{\pi}} \\
\left(\boldsymbol{E}^1\right)^{-1}\bar{\boldsymbol{y}}^1 &= \left(\boldsymbol{E}^2\right)^{-1}\bar{\boldsymbol{y}}^2 \qquad (0.1)
\end{aligned}
$$

$$(0.2)$$

Deriving Bounds on Proportions

Assumption

Wisdom of the Coders *Coder 1 and 2 have maximum validity given their agreement rate* a^{12}

Assumption

Constant Validity Assumption *Coder c's validity is constant across categories.* $\epsilon^{c} = \epsilon^{c}_{kk}$

Other structure:

$$
\begin{aligned}
\left(\boldsymbol{E}^1\right)^{-1}\bar{\boldsymbol{y}}^1 &= \bar{\boldsymbol{\pi}} \\
\left(\boldsymbol{E}^2\right)^{-1}\bar{\boldsymbol{y}}^2 &= \bar{\boldsymbol{\pi}} \\
\left(\boldsymbol{E}^1\right)^{-1}\bar{\boldsymbol{y}}^1 &= \left(\boldsymbol{E}^2\right)^{-1}\bar{\boldsymbol{y}}^2 \qquad (0.1) \\
\left(\boldsymbol{E}^1\right)^{-1}\bar{\boldsymbol{y}}^1 &\in \text{(K-1)-dimensional simplex} \qquad (0.2)
\end{aligned}
$$

Intervals for the Proportion in Each Category

Set of pairs of matrices $(\tilde{\boldsymbol{E}}^1, \tilde{\boldsymbol{E}}^2)$ that satisfy maximum average validity, constant validity, and Equations 0.1 and 0.2 into set $\mathbb{E}$.

Proposition

Intervals for the Proportion in Each Category

Set of pairs of matrices $(\tilde{\boldsymbol{E}}^1, \tilde{\boldsymbol{E}}^2)$ that satisfy maximum average validity, constant validity, and Equations 0.1 and 0.2 into set $\mathbb{E}$.

Proposition

Suppose coders have maximum average validity and constant validity.

Intervals for the Proportion in Each Category

Set of pairs of matrices $(\tilde{\boldsymbol{E}}^1, \tilde{\boldsymbol{E}}^2)$ that satisfy maximum average validity, constant validity, and Equations 0.1 and 0.2 into set $\mathbb{E}$.

Proposition

Suppose coders have maximum average validity and constant validity. Define $\bar{\pi}_k^{int}$ *as*

Intervals for the Proportion in Each Category

Set of pairs of matrices $(\tilde{\boldsymbol{E}}^1, \tilde{\boldsymbol{E}}^2)$ that satisfy maximum average validity, constant validity, and Equations 0.1 and 0.2 into set $\mathbb{E}$.

Proposition

Suppose coders have maximum average validity and constant validity. Define $\bar{\pi}_k^{int}$ as

$$\bar{\pi_k}^{int} = \left[\min_{(\tilde{E}^1, \tilde{E}^2) \in \mathbb{E}} \left(\tilde{\boldsymbol{E}}^c\right)^{-1} \bar{\boldsymbol{y}}^c|_k, \max_{(\tilde{E}^1, \tilde{E}^2) \in \mathbb{E}} \left(\tilde{\boldsymbol{E}}^c\right)^{-1} \bar{\boldsymbol{y}}^c|_k \right]$$

Intervals for the Proportion in Each Category

Set of pairs of matrices $(\tilde{\boldsymbol{E}}^1, \tilde{\boldsymbol{E}}^2)$ that satisfy maximum average validity, constant validity, and Equations 0.1 and 0.2 into set $\mathbb{E}$.

Proposition

Suppose coders have maximum average validity and constant validity. Define $\bar{\pi}_k^{int}$ as

$$\bar{\pi}_k^{int} \quad = \quad \left[\min_{(\tilde{E}^1,\tilde{E}^2)\in\mathbb{E}} \left(\tilde{\boldsymbol{E}}^c\right)^{-1} \bar{\boldsymbol{y}}^c|_k, \max_{(\tilde{E}^1,\tilde{E}^2)\in\mathbb{E}} \left(\tilde{\boldsymbol{E}}^c\right)^{-1} \bar{\boldsymbol{y}}^c|_k\right]$$

Then $\bar{\pi}_k \in \bar{\pi}_k^{int}$.

Intervals for the Proportion in Each Category

Set of pairs of matrices $(\tilde{\boldsymbol{E}}^1, \tilde{\boldsymbol{E}}^2)$ that satisfy maximum average validity, constant validity, and Equations 0.1 and 0.2 into set $\mathbb{E}$.

Proposition

Suppose coders have maximum average validity and constant validity. Define $\bar{\pi}_k^{int}$ as

$$\bar{\pi}_k^{int} \;=\; \left[\min_{(\tilde{E}^1,\tilde{E}^2)\in\mathbb{E}} \left(\tilde{\boldsymbol{E}}^c\right)^{-1} \bar{\boldsymbol{y}}^c|_k, \max_{(\tilde{E}^1,\tilde{E}^2)\in\mathbb{E}} \left(\tilde{\boldsymbol{E}}^c\right)^{-1} \bar{\boldsymbol{y}}^c|_k\right]$$

Then $\bar{\pi}_k \in \bar{\pi}_k^{int}$.

Optimization not straightforward $\rightsquigarrow$ non-linear programming algorithm

Example 1: Three Categories

Two coders: agree 70% of speeches

	Category 1	Category 2	Category 3
Truth	0.7	0.25	0.05
Naive Estimate	0.63	0.25	0.13

Example 1: Three Categories

Two coders: agree 70% of speeches

	Category 1	Category 2	Category 3
Truth	0.7	0.25	0.05
Naive Estimate	0.63	0.25	0.13
Constant Validity ($\epsilon^1, \epsilon^2 \in [0.65, 1]$)	[0.63, 0.88]	[0.00, 0.29]	[0.00,0.18]
+			

Example 1: Three Categories

Two coders: agree 70% of speeches

	Category 1	Category 2	Category 3
Truth	0.7	0.25	0.05
Naive Estimate	0.63	0.25	0.13
Constant Validity ($\epsilon^1, \epsilon^2 \in [0.65, 1]$)	[0.63, 0.88]	[0.00, 0.29]	[0.00,0.18]
+ Maximum Average Validity	[0.68, 0.77]	[0.09, 0.29]	[0.00, 0.16]

Example 1: Three Categories

Two coders: agree 70% of speeches

	Category 1	Category 2	Category 3
Truth	0.7	0.25	0.05
Naive Estimate	0.63	0.25	0.13
Constant Validity ($\epsilon^1, \epsilon^2 \in [0.65, 1]$)	[0.63, 0.88]	[0.00, 0.29]	[0.00,0.18]
+ Maximum Average Validity	[0.68, 0.77]	[0.09, 0.29]	[0.00, 0.16]
+ Structure	[0.69, 0.73]	[0.21, 0.26]	[0.02, 0.08]

Example 1: Three Categories

Two coders: agree 70% of speeches

	Category 1	Category 2	Category 3
Truth	0.7	0.25	0.05
Naive Estimate	0.63	0.25	0.13
Constant Validity ($\epsilon^1, \epsilon^2 \in [0.65, 1]$)	[0.63, 0.88]	[0.00, 0.29]	[0.00,0.18]
+ Maximum Average Validity	[0.68, 0.77]	[0.09, 0.29]	[0.00, 0.16]
+ Structure	[0.69, 0.73]	[0.21, 0.26]	[0.02, 0.08]

Naive estimate $\rightsquigarrow$ outside of bounds (Category 1 and 3)

Example 1: Three Categories

Two coders: agree 70% of speeches

	Category 1	Category 2	Category 3
Truth	0.7	0.25	0.05
Naive Estimate	0.63	0.25	0.13
Constant Validity ($\epsilon^1, \epsilon^2 \in [0.65, 1]$)	[0.63, 0.88]	[0.00, 0.29]	[0.00,0.18]
+ Maximum Average Validity	[0.68, 0.77]	[0.09, 0.29]	[0.00, 0.16]
+ Structure	[0.69, 0.73]	[0.21, 0.26]	[0.02, 0.08]

Naive estimate $\rightsquigarrow$ outside of bounds (Category 1 and 3)

High (acceptable) reliability $\neq$ unbiased inferences

Simulation Evidence

No. Coded	Bootstrap	Prop. Contained
Maximum Validity		
100	No	0.60

Simulation Evidence

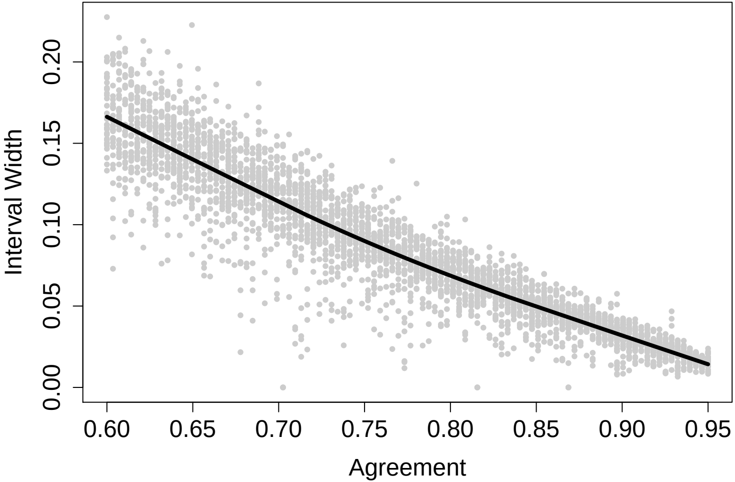
No. Coded	Bootstrap	Prop. Contained
Maximum Validity		
100	No	0.60
100	Yes	0.93

Simulation Evidence

No. Coded	Bootstrap	Prop. Contained
Maximum Validity		
100	No	0.60
100	Yes	0.93
500	No	0.93
500	Yes	1
1000	No	0.99
1000	Yes	1
10000	No	0.98
10000	No	0.99
30000	No	1
30000	No	0.99

Simulation Evidence

No. Coded	Bootstrap	Prop. Contained
Relaxing Constant Validity		
10000	No	0.86
Independent Coders		
1000	No	1
10000	No	1


Interval Width
0.00
0.05
0.10
0.15
0.20
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
Agreement

Generalize:

1) Number of coders
2) Maximum Average Validity
3) Constant Validity

Dawid-Skene (1979) Annonator Model

Computer science, NLP literature

Dawid-Skene (1979) Annonator Model

Computer science, NLP literature $\rightsquigarrow$ model annotator bias with mixture model

Dawid-Skene (1979) Annonator Model

Computer science, NLP literature $\rightsquigarrow$ model annotator bias with mixture model

$$\pi_d \sim \text{Multinomial}(1, \bar{\boldsymbol{\pi}})$$

Dawid-Skene (1979) Annonator Model

Computer science, NLP literature $\rightsquigarrow$ model annotator bias with mixture model

$$
\begin{aligned}
\pi_d &\sim \text{Multinomial}(1, \bar{\boldsymbol{\pi}}) \\
y_d^c &\sim \text{Multinomial}(1, \boldsymbol{\epsilon^c}_{\pi_d})
\end{aligned}
$$

Dawid-Skene (1979) Annonator Model

Computer science, NLP literature $\rightsquigarrow$ model annotator bias with mixture model

$$\pi_d \sim \text{Multinomial}(1, \bar{\boldsymbol{\pi}})$$
$$y_d^c \sim \text{Multinomial}(1, \boldsymbol{\epsilon^c}_{\pi_d})$$

where $\boldsymbol{\epsilon^c}_{\pi_d}$ refers to the π_d^{th} column of evaluation matrix

Dawid-Skene (1979) Annonator Model

Computer science, NLP literature $\leadsto$ model annotator bias with mixture model

$$
\begin{aligned}
\pi_d &\sim \text{Multinomial}(1, \bar{\boldsymbol{\pi}}) \\
y_d^c &\sim \text{Multinomial}(1, \boldsymbol{\epsilon^c}_{\pi_d})
\end{aligned}
$$

where $\boldsymbol{\epsilon^c}_{\pi_d}$ refers to the π_d^{th} column of evaluation matrix

Problems:

Dawid-Skene (1979) Annonator Model

Computer science, NLP literature $\rightsquigarrow$ model annotator bias with mixture model

$$\begin{aligned}
\pi_d &\sim \text{Multinomial}(1, \bar{\boldsymbol{\pi}}) \\
y_d^c &\sim \text{Multinomial}(1, \boldsymbol{\epsilon^c}_{\pi_d})
\end{aligned}$$

where $\boldsymbol{\epsilon^c}_{\pi_d}$ refers to the π_d^{th} column of evaluation matrix

Problems:

1) Sensitive to starting values $\rightsquigarrow$ bias

Dawid-Skene (1979) Annonator Model

Computer science, NLP literature $\leadsto$ model annotator bias with mixture model

$$\begin{aligned} \pi_d &\sim \text{Multinomial}(1, \bar{\boldsymbol{\pi}}) \\ y_d^c &\sim \text{Multinomial}(1, \boldsymbol{\epsilon^c}_{\pi_d}) \end{aligned}$$

where $\boldsymbol{\epsilon^c}_{\pi_d}$ refers to the π_d^{th} column of evaluation matrix

Problems:

1) Sensitive to starting values $\leadsto$ bias
2) Individual document labels $\leadsto$ sensitive to starting value

Dawid-Skene (1979) Annonator Model

Computer science, NLP literature $\rightsquigarrow$ model annotator bias with mixture model

$$\begin{aligned} \pi_d &\sim \text{Multinomial}(1, \bar{\boldsymbol{\pi}}) \\ y_d^c &\sim \text{Multinomial}(1, \boldsymbol{\epsilon^c}_{\pi_d}) \end{aligned}$$

where $\boldsymbol{\epsilon^c}_{\pi_d}$ refers to the π_d^{th} column of evaluation matrix

Problems:

1) Sensitive to starting values $\rightsquigarrow$ bias
2) Individual document labels $\rightsquigarrow$ sensitive to starting value
3) Systematic bias in inferred proportions

Criticism and Vitriol (Grimmer, King, and Superti 2015a)

Criticism and Vitriol (Grimmer, King, and Superti 2015a)

Taunting (Vitrol): attack other party's (or member's) competency (Valence)

Criticism and Vitriol (Grimmer, King, and Superti 2015a)

Taunting (Vitrol): attack other party's (or member's) competency (Valence)
Taunting: explict, public, and negative attacks

Criticism and Vitriol (Grimmer, King, and Superti 2015a)

Taunting (Vitrol): attack other party's (or member's) competency (Valence)
Taunting: explict, public, and negative attacks

- Sample 30,000 Senate Floor Speeches $\rightsquigarrow$ Taunting, Other Categories

Criticism and Vitriol (Grimmer, King, and Superti 2015a)

Taunting (Vitrol): attack other party's (or member's) competency (Valence)
Taunting: explict, public, and negative attacks

- Sample 30,000 Senate Floor Speeches $\rightsquigarrow$ Taunting, Other Categories
- 10% of speeches double coded, random pair of coders
- Relative high agreement rate ($\approx$ 85%), with face validity

Criticism and Vitriol (Grimmer, King, and Superti 2015a)

Taunting (Vitrol): attack other party's (or member's) competency (Valence)
Taunting: explict, public, and negative attacks

- Sample 30,000 Senate Floor Speeches $\rightsquigarrow$ Taunting, Other Categories
- 10% of speeches double coded, random pair of coders
- Relative high agreement rate ($\approx$ 85%), with face validity
- Interested in average rate senators taunt in their floor speeches

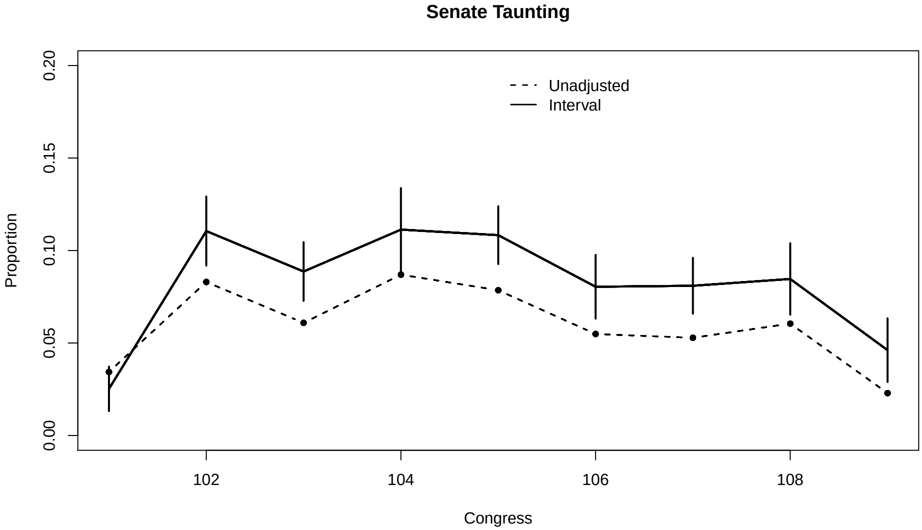
Criticism and Vitriol (Grimmer, King, and Superti 2015a)

Taunting (Vitrol): attack other party's (or member's) competency (Valence)
Taunting: explict, public, and negative attacks

- Sample 30,000 Senate Floor Speeches $\rightsquigarrow$ Taunting, Other Categories
- 10% of speeches double coded, random pair of coders
- Relative high agreement rate ($\approx$ 85%), with face validity
- Interested in average rate senators taunt in their floor speeches

Use extensions to apply algorithm to estimate Congress-to-Congress changes in taunting rate with non-overlapping coders

Partisan Taunting



The Problem of Intercoder Reliability

Our Solution:

- Intervals that contain truth with probabilty 1
- Extensions (in the paper) include:
  - Bounds on agreement with alloyed gold standard for machine learning methods
  - Multiple coders (wisdom of crowds results)
  - Proportions as inputs to other models
- Extensions (outside paper) include:
  - Analysis of Computer Science prediction contests

The Problem of Intercoder Reliability

The Problem of Intercoder Reliability

Coder Error $\rightsquigarrow$ Bias

The Problem of Intercoder Reliability

Coder Error $\rightsquigarrow$ Bias

Coder Error $\rightsquigarrow$ Method to Address Bias