Supervised Methods for Classifying and Scaling Texts

MY560 Workshop in Advanced Quantitative Analysis

Kenneth Benoit

May 28, 2013

Today's Road Map

10:00–10:55 Introduction to the Naive Bayes Classifier

11:05–12:00 *Break*

11:05–12:00 Using Classification Posteriors for Scaling Texts

12:00–14:00 *Lunch Break*

14:00–16:00 Lab session: Classifying Text Using Wordstat

INTRODUCTION TO NAIVE BAYES

Prior probabilities and updating

A test is devised to automatically flag racist news stories.

- 1% of news stories in general have racist messages
- 80% of racist news stories will be flagged by the test
- 10% of non-racist stories will also be flagged

Prior probabilities and updating

A test is devised to automatically flag racist news stories.

- 1% of news stories in general have racist messages
- 80% of racist news stories will be flagged by the test
- 10% of non-racist stories will also be flagged

We run the test on a new news story, and it is *flagged as* racist.

Prior probabilities and updating

A test is devised to automatically flag racist news stories.

- 1% of news stories in general have racist messages
- 80% of racist news stories will be flagged by the test
- 10% of non-racist stories will also be flagged

We run the test on a new news story, and it is *flagged as* racist.

Question: What is probability that the story is *actually* racist?

Prior probabilities and updating

A test is devised to automatically flag racist news stories.

- 1% of news stories in general have racist messages
- 80% of racist news stories will be flagged by the test
- 10% of non-racist stories will also be flagged

We run the test on a new news story, and it is *flagged as* racist.

Question: What is probability that the story is *actually* racist?

Any guesses?

Prior probabilities and updating

- What about without the test?

Prior probabilities and updating

- What about without the test?
  - Imagine we run 1,000 news stories through the test

Prior probabilities and updating

- What about without the test?
  - Imagine we run 1,000 news stories through the test
  - We expect that 10 will be racist

Prior probabilities and updating

- What about without the test?
  - Imagine we run 1,000 news stories through the test
  - We expect that 10 will be racist
- With the test, we expect:

Prior probabilities and updating

- What about without the test?
  - Imagine we run 1,000 news stories through the test
  - We expect that 10 will be racist
- With the test, we expect:
  - Of the 10 found to be racist,

Prior probabilities and updating

- What about without the test?
  - Imagine we run 1,000 news stories through the test
  - We expect that 10 will be racist
- With the test, we expect:
  - Of the 10 found to be racist, 8 should be flagged as racist

Prior probabilities and updating

- What about without the test?
  - Imagine we run 1,000 news stories through the test
  - We expect that 10 will be racist
- With the test, we expect:
  - Of the 10 found to be racist, 8 should be flagged as racist
  - Of the 990 non-racist stories,

Prior probabilities and updating

- What about without the test?
  - Imagine we run 1,000 news stories through the test
  - We expect that 10 will be racist
- With the test, we expect:
  - Of the 10 found to be racist, 8 should be flagged as racist
  - Of the 990 non-racist stories, 99 will be wrongly flagged as racist

Prior probabilities and updating

- What about without the test?
  - Imagine we run 1,000 news stories through the test
  - We expect that 10 will be racist
- With the test, we expect:
  - Of the 10 found to be racist, 8 should be flagged as racist
  - Of the 990 non-racist stories, 99 will be wrongly flagged as racist
  - That's a total of 107 stories flagged as racist

Prior probabilities and updating

- What about without the test?
  - Imagine we run 1,000 news stories through the test
  - We expect that 10 will be racist
- With the test, we expect:
  - Of the 10 found to be racist, 8 should be flagged as racist
  - Of the 990 non-racist stories, 99 will be wrongly flagged as racist
  - That's a total of 107 stories flagged as racist
- So: the updated probability of a story being racist, conditional on being flagged as racist, is

Prior probabilities and updating

- What about without the test?
  - Imagine we run 1,000 news stories through the test
  - We expect that 10 will be racist
- With the test, we expect:
  - Of the 10 found to be racist, 8 should be flagged as racist
  - Of the 990 non-racist stories, 99 will be wrongly flagged as racist
  - That's a total of 107 stories flagged as racist
- So: the updated probability of a story being racist, conditional on being flagged as racist, is $\frac{8}{107} = 0.075$
- The *prior* probability of 0.01 is updated to only 0.075 by the positive test result

This is an example of Bayes' Rule:

$$P(R = 1 | T = 1) = \frac{P(T=1|R=1)P(R=1)}{P(T=1)}$$

Multinomial Bayes model of Class given a Word

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

At the word level, Bayes Theorem tells us that:

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

Multinomial Bayes model of Class given a Word

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

At the word level, Bayes Theorem tells us that:

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

For two classes, this can be expressed as

$$= \frac{P(w_j|c_k)P(c_j)}{P(w_j|c_k)P(c_k) + P(w_j|c_{\neg k})P(c_{\neg k})} \tag{1}$$

Classification as a goal

- Machine learning focuses on identifying classes (classification), while social science is typically interested in locating things on latent traits (scaling)
- One of the simplest and most robust classification methods is the "Naive Bayes" (NB) classifier, built on a Bayesian probability model
- The class predictions for a collection of words from NB are great for classification, but useless for scaling
- But intermediate steps from NB turn out to be excellent for scaling purposes, and identical to Laver, Benoit and Garry's "Wordscores"
- Applying lessons from machine to learning to supervised scaling, we can
  - Apply classification methods to scaling
  - improve it using lessons from machine learning

Supervised v. unsupervised methods compared

- The goal (in text analysis) is to differentiate *documents* from one another, treating them as "bags of words"
- Different approaches:
  - *Supervised methods* require a training set that exmplify constrasting classes, identified by the researcher
  - *Unsupervised methods* scale documents based on patterns of similarity from the term-document matrix, without requiring a training step

Supervised v. unsupervised methods compared

- The goal (in text analysis) is to differentiate *documents* from one another, treating them as "bags of words"
- Different approaches:
  - *Supervised methods* require a training set that exmplify constrasting classes, identified by the researcher
  - *Unsupervised methods* scale documents based on patterns of similarity from the term-document matrix, without requiring a training step
- Relative advantage of supervised methods:
  You already know the dimension being scaled, because you set it in the training stage

Supervised v. unsupervised methods compared

- The goal (in text analysis) is to differentiate *documents* from one another, treating them as "bags of words"
- Different approaches:
  - *Supervised methods* require a training set that exmplify constrasting classes, identified by the researcher
  - *Unsupervised methods* scale documents based on patterns of similarity from the term-document matrix, without requiring a training step
- Relative advantage of supervised methods:
  You already know the dimension being scaled, because you set it in the training stage
- Relative disadvantage of supervised methods:
  You *must* already know the dimension being scaled, because you have to feed it good sample documents in the training stage

Supervised v. unsupervised methods: Examples

- General examples:
  - Supervised: Naive Bayes, k-Nearest Neighbor, Support Vector Machines (SVM)
  - Unsupervised: correspondence analysis, IRT models, factor analytic approaches

Supervised v. unsupervised methods: Examples

- General examples:
  - Supervised: Naive Bayes, k-Nearest Neighbor, Support Vector Machines (SVM)
  - Unsupervised: correspondence analysis, IRT models, factor analytic approaches
- Political science applications
  - Supervised: Wordscores (LBG 2003); SVMs (Yu, Kaufman and Diermeier 2008); Naive Bayes (Evans et al 2007)
  - Unsupervised "Wordfish" (Slapin and Proksch 2008); Correspondence analysis (Schonhardt-Bailey 2008); two-dimensional IRT (Monroe and Maeda 2004)

Focus today

- The focus today will be on Naive Bayes
- We will also cover the Laver, Benoit and Garry (2003) “Wordscores” scaling method

Multinomial Bayes model of Class given a Word

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

At the word level, Bayes Theorem tells us that:

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

For two classes, this can be expressed as

$$= \frac{P(w_j|c_k)P(c_j)}{P(w_j|c_k)P(c_k) + P(w_j|c_{\neg k})P(c_{\neg k})} \qquad (2)$$

Moving to the document level

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

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

Moving to the document level

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

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

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

Multinomial Bayes model of Class given a Word

Class-conditional word likelihoods

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

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

Multinomial Bayes model of Class given a Word

Word probabilities

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

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

Multinomial Bayes model of Class given a Word

Class prior probabilities

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

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

Multinomial Bayes model of Class given a Word
Class posterior probabilities

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

- This represents the posterior probability of membership in class k for word j
- Under *certain conditions*, this is identical to what LBG (2003) called P_{wr}
- Under those conditions, the LBG "wordscore" is the linear difference between $P(c_k|w_j)$ and $P(c_{\neg k}|w_j)$

Naive Bayes Classification Example

(From Manning, Raghavan and Schütze, *Introduction to Information Retrieval*)

► **Table 13.1** Data for parameter estimation examples.

	docID	words in document	in $c = China$?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

Naive Bayes Classification Example

Example 13.1: For the example in Table 13.1, the multinomial parameters we need to classify the test document are the priors $\hat{P}(c) = 3/4$ and $\hat{P}(\overline{c}) = 1/4$ and the following conditional probabilities:

$$\begin{aligned}
\hat{P}(\textsf{Chinese}|c) &= (5+1)/(8+6) = 6/14 = 3/7 \\
\hat{P}(\textsf{Tokyo}|c) = \hat{P}(\textsf{Japan}|c) &= (0+1)/(8+6) = 1/14 \\
\hat{P}(\textsf{Chinese}|\overline{c}) &= (1+1)/(3+6) = 2/9 \\
\hat{P}(\textsf{Tokyo}|\overline{c}) = \hat{P}(\textsf{Japan}|\overline{c}) &= (1+1)/(3+6) = 2/9
\end{aligned}$$

The denominators are $(8+6)$ and $(3+6)$ because the lengths of $text_c$ and $text_{\overline{c}}$ are 8 and 3, respectively, and because the constant B in Equation (13.7) is 6 as the vocabulary consists of six terms.

We then get:

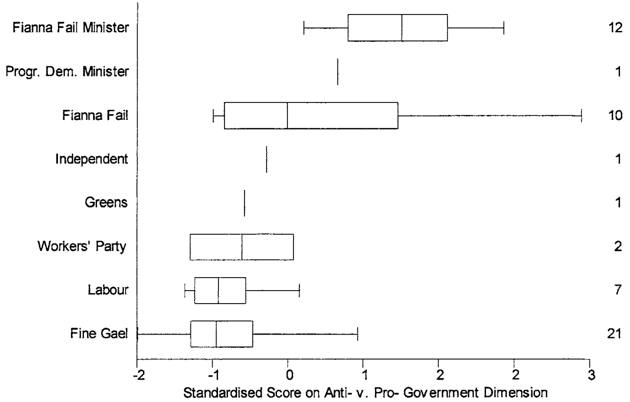
$$\begin{aligned}
\hat{P}(c|d_5) &\propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003. \\
\hat{P}(\overline{c}|d_5) &\propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001.
\end{aligned}$$

Thus, the classifier assigns the test document to c = *China*. The reason for this classification decision is that the three occurrences of the positive indicator Chinese in d_5 outweigh the occurrences of the two negative indicators Japan and Tokyo.

SCALING TEXTS

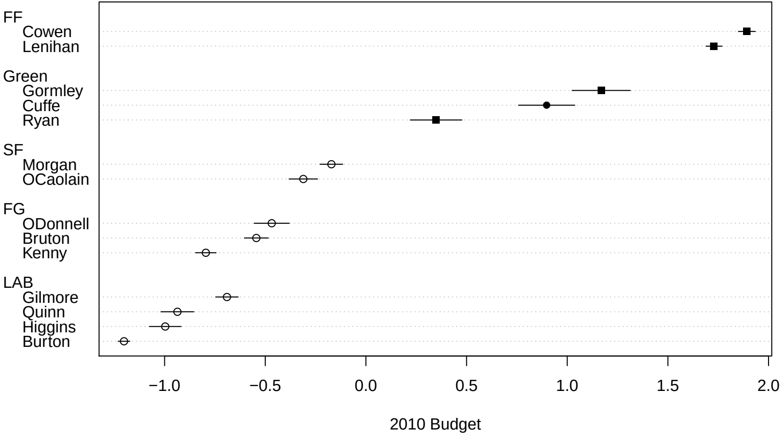
No confidence debate speeches (Wordscores)

FIGURE 3. Box Plot of Standardized Scores of Speakers in 1991 Confidence Debate on "Pro- versus Antigovernment" Dimension, by Category of Legislator

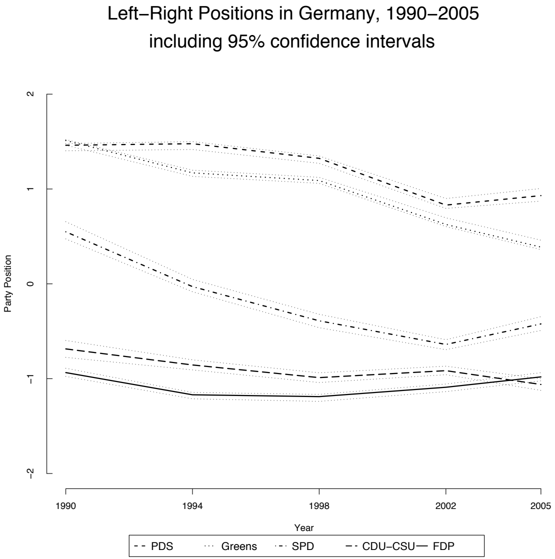


(from Benoit and Laver, *Irish Political Studies* 2002)

Government v. Opposition in Irish Budget Debate (2010)



Party Manifestos: Poisson scaling



(from Slapin and Proksch, *AJPS* 2008)

First impressions?

- How many dimensions?

First impressions?

- How many dimensions?
- What is the scale?

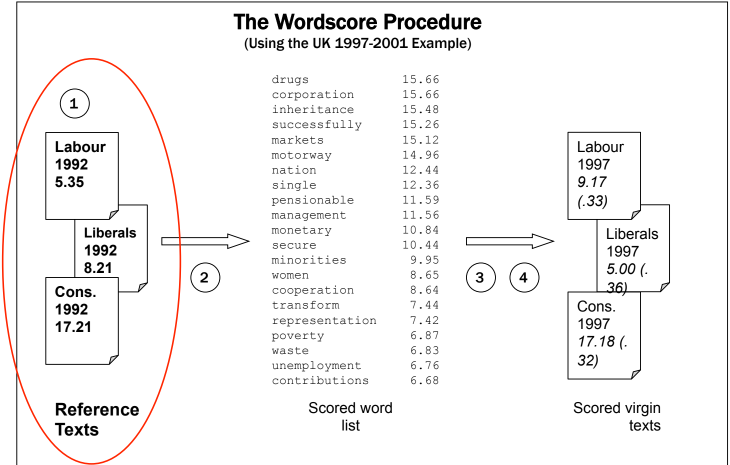
First impressions?

- How many dimensions?
- What is the scale?
- Uncertainty?

Wordscores conceptually

- Two sets of texts
  - Reference texts: texts about which we know something (a scalar dimensional score)
  - Virgin texts: texts about which we know nothing (but whose dimensional score wed like to know)
- These are analogous to a "training set" and a "test set" in classification
- Basic procedure:
  1. Analyze reference texts to obtain word scores
  2. Use word scores to score virgin texts

Wordscores Procedure



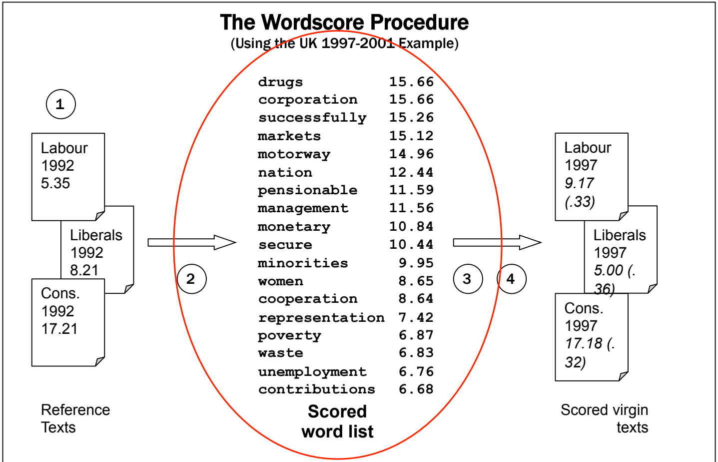
Step 1: Obtain reference texts with a priori known positions (`setref`)

Step 2: Generate word scores from reference texts (`wordscore`)

Step 3: Score each virgin text using word scores (`textscore`)

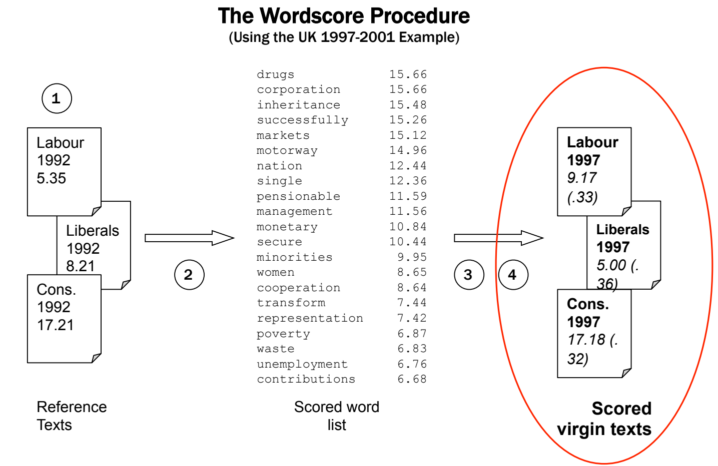
Step 4: (optional) Transform virgin text scores to original metric

Wordscores Procedure



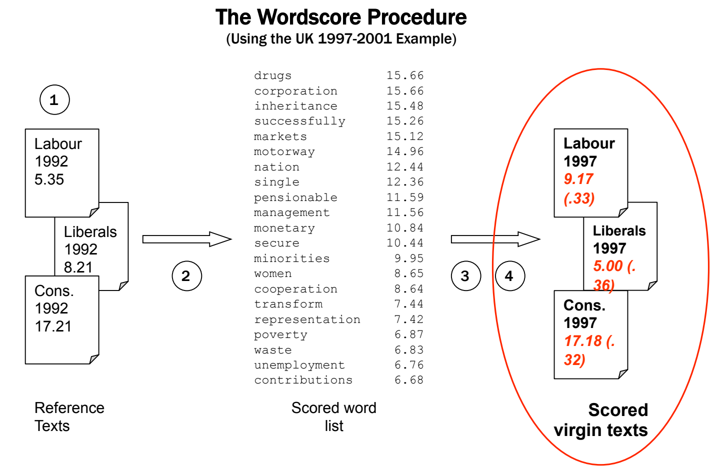
Step 1: Obtain reference texts with a priori known positions (`setref`)
Step 2: Generate word scores from reference texts (`wordscore`)
Step 3: Score each virgin text using word scores (`textscore`)
Step 4: (optional) Transform virgin text scores to original metric

Wordscores Procedure



Step 1: Obtain reference texts with a priori known positions (`setref`)
Step 2: Generate word scores from reference texts (`wordscore`)
Step 3: Score each virgin text using word scores (`textscore`)
Step 4: (optional) Transform virgin text scores to original metric

Wordscores Procedure



Step 1: Obtain reference texts with a priori known positions (`setref`)
Step 2: Generate word scores from reference texts (`wordscore`)
Step 3: Score each virgin text using word scores (`textscore`)
Step 4: (optional) Transform virgin text scores to original metric

Wordscores mathematically: Reference texts

- Start with a set of I *reference* texts, represented by an $I \times J$ document-term frequency matrix C_{ij}, where i indexes the document and j indexes the J total word types

Wordscores mathematically: Reference texts

- Start with a set of I *reference* texts, represented by an $I \times J$ document-term frequency matrix C_{ij}, where i indexes the document and j indexes the J total word types

Wordscores mathematically: Reference texts

- Start with a set of I *reference* texts, represented by an $I \times J$ document-term frequency matrix C_{ij}, where i indexes the document and j indexes the J total word types
- Each text will have an associated "score" a_i, which is a single number locating this text on a single dimension of difference

Wordscores mathematically: Reference texts

- Start with a set of I *reference* texts, represented by an $I \times J$ document-term frequency matrix C_{ij}, where i indexes the document and j indexes the J total word types
- Each text will have an associated "score" a_i, which is a single number locating this text on a single dimension of difference
  - This can be on a scale metric, such as 1–20
  - Can use arbitrary endpoints, such as -1, 1

Wordscores mathematically: Reference texts

- Start with a set of I *reference* texts, represented by an $I \times J$ document-term frequency matrix C_{ij}, where i indexes the document and j indexes the J total word types
- Each text will have an associated "score" a_i, which is a single number locating this text on a single dimension of difference
  - This can be on a scale metric, such as 1–20
  - Can use arbitrary endpoints, such as -1, 1
- We *normalize* the document-term frequency matrix within each document by converting C_{ij} into a *relative* document-term frequency matrix (within document), by dividing C_{ij} by its word total marginals:

$$F_{ij} = \frac{C_{ij}}{C_{i\cdot}} \tag{3}$$

where $C_{i\cdot} = \sum_{j=1}^{J} C_{ij}$

Wordscores mathematically: Word scores

- Compute an $I \times J$ matrix of relative document probabilities P_{ij} for each word in each reference text, as

$$P_{ij} = \frac{F_{ij}}{\sum_{i=1}^{I} F_{ij}} \qquad (4)$$

Wordscores mathematically: Word scores

- Compute an $I \times J$ matrix of relative document probabilities P_{ij} for each word in each reference text, as

$$P_{ij} = \frac{F_{ij}}{\sum_{i=1}^{I} F_{ij}} \tag{4}$$

- This tells us the probability that given the observation of a specific word j, that we are reading a text of a certain reference document i

Wordscores mathematically: Word scores (example)

- Assume we have two reference texts, A and B
- The word "choice" is used 10 times per 1,000 words in Text A and 30 times per 1,000 words in Text B
- So $F_{i\ \text{"choice"}} = \{.010, .030\}$
- If we know only that we are reading the word choice in one of the two reference texts, then probability is 0.25 that we are reading Text A, and 0.75 that we are reading Text B

$$P(A|\text{"choice"}) = \frac{.1}{(.1 + .3)} = 0.25$$

$$P(B|\text{"choice"}) = \frac{.3}{(.1 + .3)} = 0.75$$

Wordscores mathematically: Word scores

- Compute a J-length "score" vector S for each word j as the average of each document i's scores a_i, weighted by each word's P_{ij}:

$$S_j = \sum_{i=1}^{I} a_i P_{ij} \tag{5}$$

Wordscores mathematically: Word scores

- Compute a J-length "score" vector S for each word j as the average of each document i's scores a_i, weighted by each word's P_{ij}:

$$S_j = \sum_{i=1}^{I} a_i P_{ij} \tag{5}$$

- In matrix algebra, $\underset{1\times J}{S} = \underset{1\times I}{a} \cdot \underset{I\times J}{P}$
- This procedure will yield a single "score" for every word that reflects the balance of the scores of the reference documents, weighted by the relative document frequency of its normalized term frequency

Wordscores mathematically: Word scores

- Continuing with our example:
  - We "know" (from independent sources) that Reference Text A has a position of -1.0, and Reference Text B has a position of $+1.0$
  - The score of the word choice is then $0.25(-1.0) + 0.75(1.0) = -0.25 + 0.75 = +0.50$

Wordscores mathematically: Scoring "virgin" texts

- Here the objective is to obtain a single score for any new text, relative to the reference texts

Wordscores mathematically: Scoring “virgin” texts

- Here the objective is to obtain a single score for any new text, relative to the reference texts
- We do this by taking the mean of the scores of its words, weighted by their term frequency

Wordscores mathematically: Scoring "virgin" texts

- Here the objective is to obtain a single score for any new text, relative to the reference texts
- We do this by taking the mean of the scores of its words, weighted by their term frequency
- So the score v_k of a virgin document k consisting of the j word types is:

$$v_k = \sum_j (F_{kj} \cdot s_j) \quad (6)$$

where $F_{kj} = \frac{C_{kj}}{C_{k\cdot}}$ as in the reference document relative word frequencies

Wordscores mathematically: Scoring "virgin" texts

- Here the objective is to obtain a single score for any new text, relative to the reference texts
- We do this by taking the mean of the scores of its words, weighted by their term frequency
- So the score v_k of a virgin document k consisting of the j word types is:

$$v_k = \sum_j (F_{kj} \cdot s_j) \tag{6}$$

  where $F_{kj} = \frac{C_{kj}}{C_{k\cdot}}$ as in the reference document relative word frequencies
- Note that new words outside of the set J may appear in the K virgin documents — these are simply ignored (because we have no information on their scores)

Wordscores mathematically: Scoring “virgin” texts

- Here the objective is to obtain a single score for any new text, relative to the reference texts
- We do this by taking the mean of the scores of its words, weighted by their term frequency
- So the score v_k of a virgin document k consisting of the j word types is:

$$v_k = \sum_j (F_{kj} \cdot s_j) \qquad (6)$$

  where $F_{kj} = \frac{C_{kj}}{C_{k\cdot}}$ as in the reference document relative word frequencies
- Note that new words outside of the set J may appear in the K virgin documents — these are simply ignored (because we have no information on their scores)
- Note also that nothing prohibits reference documents from also being scored as virgin documents

Wordscores mathematically: Rescaling raw text scores

- Because of overlapping or non-discriminating words, the raw text scores will be dragged to the interior of the reference scores (we will see this shortly in the results)

Wordscores mathematically: Rescaling raw text scores

- Because of overlapping or non-discriminating words, the raw text scores will be dragged to the interior of the reference scores (we will see this shortly in the results)
- Some procedures can be applied to rescale them, either to a unit normal metric or to a more “natural” metric

Wordscores mathematically: Rescaling raw text scores

- Because of overlapping or non-discriminating words, the raw text scores will be dragged to the interior of the reference scores (we will see this shortly in the results)
- Some procedures can be applied to rescale them, either to a unit normal metric or to a more "natural" metric
- Martin and Vanberg (2008) have proposed alternatives to the LBG (2003) rescaling
- Not a terribly important issue

Multinomial Bayes model of Class given a Word

Class posterior probabilities

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

- This represents the posterior probability of membership in class k for word j
- Under *certain conditions*, this is identical to what LBG (2003) called P_{wr}
- Under those conditions, the LBG "wordscore" is the linear difference between $P(c_k|w_j)$ and $P(c_{\neg k}|w_j)$

“Certain conditions”

“Certain conditions”

- The LBG approach required the identification not only of texts for each training class, but also “reference” scores attached to each training class
- Consider two “reference” scores s_1 and s_2 attached to two classes $k = 1$ and $k = 2$. Taking P_1 as the posterior $P(k = 1|w = j)$ and P_2 as $P(k = 2|w = j)$, A generalised score s_j^* for the word j is then

$$
\begin{aligned}
s_j^* &= s_1 P_1 + s_2 P_2 \\
&= s_1 P_1 + s_2(1 - P_1) \\
&= s_1 P_1 + s_2 - s_2 P_1) \\
&= P_1(s_1 - s_2) + s_2
\end{aligned}
$$

From Classification to Scaling

- The class predictions for a collection of words from NB can be adapted to scaling
- The intermediate steps from NB turn out to be excellent for scaling purposes, and identical to Laver, Benoit and Garry's "Wordscores"
- There are certain things from machine learning that ought to be adopted when classification methods are used for scaling
  - Feature selection
  - Stemming/pre-processing

Multinomial Bayes model of Class given a Word

Class posterior probabilities

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

- This represents the posterior probability of membership in class k for word j
- Under *certain conditions*, this is identical to what LBG (2003) called P_{wr}
- Under those conditions, the LBG "wordscore" is the linear difference between $P(c_k|w_j)$ and $P(c_{\neg k}|w_j)$

“Certain conditions”

“Certain conditions”

- The LBG approach required the identification not only of texts for each training class, but also “reference” scores attached to each training class
- Consider two “reference” scores s_1 and s_2 attached to two classes $k = 1$ and $k = 2$. Taking P_1 as the posterior $P(k = 1|w = j)$ and P_2 as $P(k = 2|w = j)$, A generalised score s_j^* for the word j is then

$$
\begin{aligned}
s_j^* &= s_1 P_1 + s_2 P_2 \\
&= s_1 P_1 + s_2(1 - P_1) \\
&= s_1 P_1 + s_2 - s_2 P_1) \\
&= P_1(s_1 - s_2) + s_2
\end{aligned}
$$

"Certain conditions": More than two reference classes

- For more than two reference classes, if the reference scores are ordered such that $s_1 < s_2 < \cdots < s_K$, then

$$
\begin{aligned}
s_j^* &= s_1 P_1 + s_2 P_2 + \cdots + s_K P_K \\
&= s_1 P_1 + s_2 P_2 + \cdots + s_K (1 - \sum_{k=1}^{K-1} P_k) \\
&= \sum_{k=1}^{K-1} P_i (s_k - s_K) + s_I
\end{aligned}
$$

A simpler formulation:
Use reference scores such that $s_1 = -1.0, s_K = 1.0$

- From above equations, it should be clear that any set of reference scores can be linearly rescaled to endpoints of $-1.0, 1.0$
- This simplifies the "simple word score"

$$s_j^* = (1 - 2P_1) + \sum_{k=2}^{K-1} P_k(s_k - 1)$$

- which simplifies with just two reference classes to:

$$s_j^* = 1 - 2P_1$$

Implications

- LBG's "word scores" come from a linear combination of class posterior probabilities from a Bayesian model of class conditional on words

Implications

- LBG's "word scores" come from a linear combination of class posterior probabilities from a Bayesian model of class conditional on words
- We might as well always anchor reference scores at $-1.0, 1.0$

Implications

- LBG's "word scores" come from a linear combination of class posterior probabilities from a Bayesian model of class conditional on words
- We might as well always anchor reference scores at $-1.0, 1.0$
- There is a special role for reference classes in between $-1.0, 1.0$, as they balance between "pure" classes — more in a moment

Implications

- LBG's "word scores" come from a linear combination of class posterior probabilities from a Bayesian model of class conditional on words
- We might as well always anchor reference scores at $-1.0, 1.0$
- There is a special role for reference classes in between $-1.0, 1.0$, as they balance between "pure" classes — more in a moment
- There are alternative scaling models, such that used in Beauchamp's (2012) "Bayesscore", which is simply the difference in logged class posteriors at the word level. For $s_1 = -1.0$, $s_2 = 1.0$,

$$
\begin{aligned}
s_j^B &= -\log P_1 + \log P_2 \\
&= \log \frac{1 - P_1}{P_1}
\end{aligned}
$$

Moving to the document level

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

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

Moving to the document level

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

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

- So we *could* consider a document-level relative score, e.g. $1 - 2P(c_1|d)$ (for a two-class problem)
- But this turns out to be *useless*, since the predictions of class are highly separated

Moving to the document level

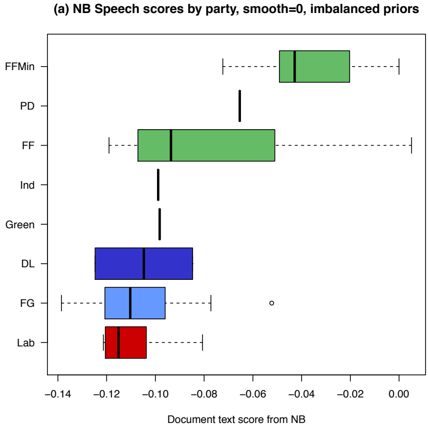
- A better solution is to score a test document as the arithmetic mean of the scores of its words
- This is exactly the solution proposed by LBG (2003)

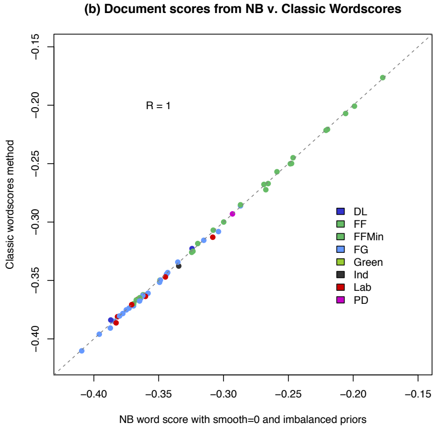
Moving to the document level

- A better solution is to score a test document as the arithmetic mean of the scores of its words
- This is exactly the solution proposed by LBG (2003)
- Beauchamp (2012) proposes a "Bayesscore" which is the arithmetic mean of the log difference word scores in a document – which yields extremely similar results

And now for some demonstrations with data...

Application 1: Dail speeches from LBG (2003)

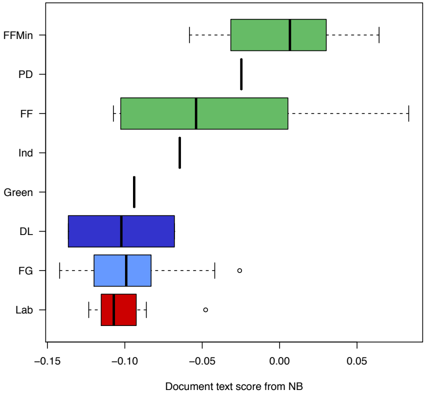




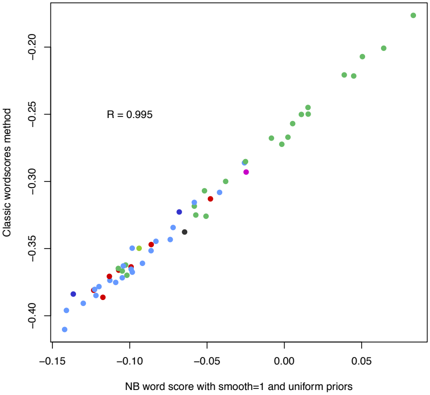
- three reference classes (Opposition, Opposition, Government) at {-1, -1, 1}
- no smoothing

Application 1: Dail speeches from LBG (2003)

(c) NB Speech scores by party, smooth=1, uniform class priors

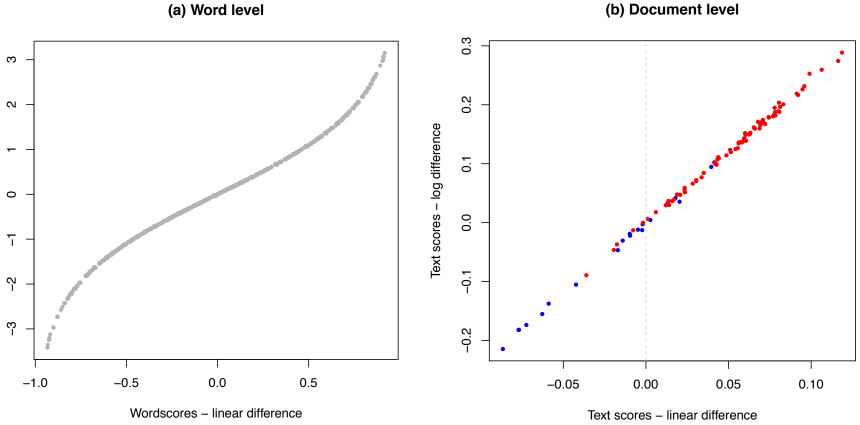


(d) Document scores from NB v. Classic Wordscores



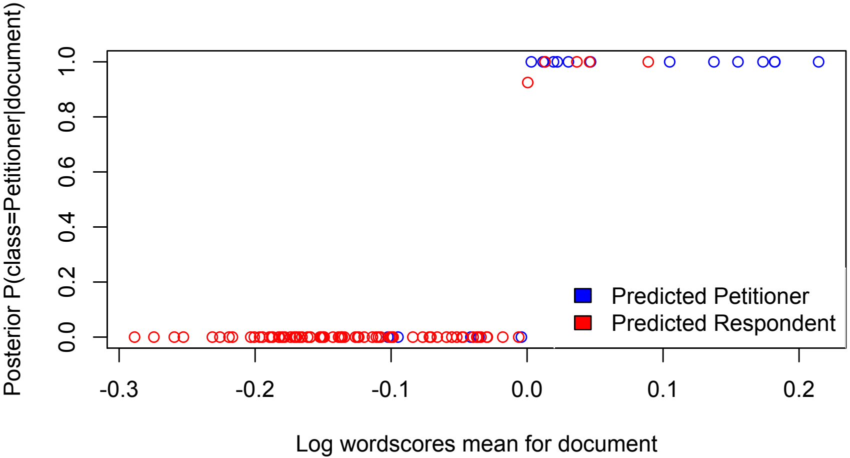
- two reference classes (Opposition+Opposition, Government) at {-1, 1}
- Laplace smoothing

Application 2: Classifying legal briefs (Evans et al 2007) Wordscores v. Bayesscore

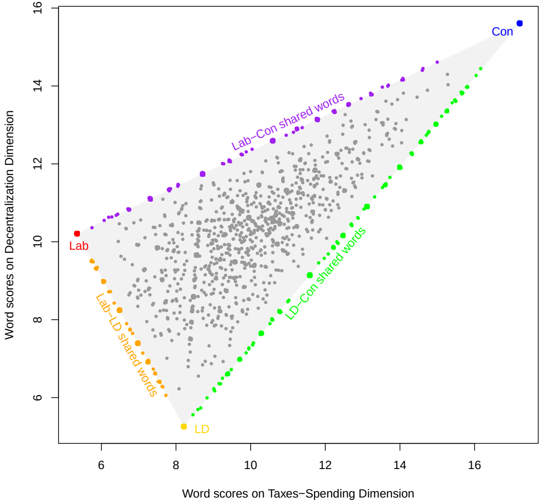


- Training set: Petitioner and Respondent litigant briefs from *Grutter/Gratz v. Bollinger* (a U.S. Supreme Court case)
- Test set: 98 amicus curiae briefs (whose P or R class is known)

Application 2: Classifying legal briefs (Evans et al 2007)
Posterior class prediction from NB versus log wordscores



Application 3: LBG's British manifestos
More than two reference classes



- x-axis: Reference scores of $\{5.35, 8.21, 17.21\}$ for Lab, LD, Conservatives
- y-axis: Reference scores of $\{10.21, 5.26, 15.61\}$

Application 4: Back to Evans et al (2007) for some Feature Selection

Machine learning commonly selects additional or deselects existing *features*:

- select (top 200) bi-grams and (top 50) trigrams, e.g. "capital punishment"
- exclude (top 200) stop words, e.g. "the", "and", ...
- count only binary word occurrence (Bernoulli NB)
- experiment with smoothing

For testing we returned to the *amicus curiae* briefs of Evans et al (2007)

Application 4: Back to Evans et al (2007) for some Feature Selection: Bigram example

Summary Judgment	Silver Rudolph	Sheila Foster
prima facie	COLLECTED WORKS	Strict Scrutiny
Jim Crow	waiting lists	Trail Transp
stare decisis	Academic Freedom	Van Alstyne
Church Missouri	General Bldg	Writings Fehrenbacher
Gerhard Casper	Goodwin Liu	boot camp
Juan Williams	Kurland Gerhard	dated April
LANDMARK BRIEFS	Lee Appearance	extracurricular activities
Lutheran Church	Missouri Synod	financial aid
Narrowly Tailored	Planned Parenthood	scored sections

Top bigrams detected using the mutual information measure

Application 4: Back to Evans et al (2007) for some Feature Selection: Classification results

				Method			
	Parameters			Wordscores		Naive Bayes Scal	
Smoothing	Stopwords	Bigrams	Distribution	Accuracy	*F*1	Accuracy	*F*1
No	No	No	Multi	0.897	0.836	-	-
No	No	No	Bern	0.459	0.647	-	-
Add-1	No	No	Multi	0.897	0.836	0.897	0.83
Add-1	No	No	Bern	-	-	0.489	0.63
Add-1	Yes	No	Multi	0.897	0.843	0.918	0.86
Add-1	Yes	No	Bern	-	-	0.500	0.62
Add-1	Yes	Yes	Multi	0.887	0.810	0.897	0.83
Add-1	Yes	Yes	Bern	-	-	0.785	0.71

Relative performance of NB and Wordscores as classifiers, given different feature selection.

(*F1* score is the harmonic mean of average precision and recall)

Conclusions

- The venerable LBG 2003 wordscores method is based on an underlying Bayesian probability model

Conclusions

- The venerable LBG 2003 wordscores method is based on an underlying Bayesian probability model
- Naive Bayes class prediction is useless for scaling, but Bayesian posterior scaling (using arithmetic means) is (also) useful for classification

Conclusions

- The venerable LBG 2003 wordscores method is based on an underlying Bayesian probability model
- Naive Bayes class prediction is useless for scaling, but Bayesian posterior scaling (using arithmetic means) is (also) useful for classification
- Always use $-1, 1$ reference scores

Conclusions

- The venerable LBG 2003 wordscores method is based on an underlying Bayesian probability model
- Naive Bayes class prediction is useless for scaling, but Bayesian posterior scaling (using arithmetic means) is (also) useful for classification
- Always use $-1, 1$ reference scores
- Two class training sets are preferred, since middle classes only combine extreme classes

Conclusions

- The venerable LBG 2003 wordscores method is based on an underlying Bayesian probability model
- Naive Bayes class prediction is useless for scaling, but Bayesian posterior scaling (using arithmetic means) is (also) useful for classification
- Always use $-1, 1$ reference scores
- Two class training sets are preferred, since middle classes only combine extreme classes
- Use uniform priors – this implies aggregating training documents by class

Conclusions

- The venerable LBG 2003 wordscores method is based on an underlying Bayesian probability model
- Naive Bayes class prediction is useless for scaling, but Bayesian posterior scaling (using arithmetic means) is (also) useful for classification
- Always use $-1, 1$ reference scores
- Two class training sets are preferred, since middle classes only combine extreme classes
- Use uniform priors – this implies aggregating training documents by class
- No knockout results from feature selection so far, implying just using the unfiltered texts seems to be OK for supervised methods

Computing confidence intervals

- The score v_k of any text represents a weighted mean

Computing confidence intervals

- The score v_k of any text represents a weighted mean
- LBG (2003) used this logic to develop a standard error of this mean using a *weighted variance* of the scores in the virgin text

Computing confidence intervals

- The score v_k of any text represents a weighted mean
- LBG (2003) used this logic to develop a standard error of this mean using a *weighted variance* of the scores in the virgin text
- Given some assumptions about the scores being fixed (and the words being conditionally independent), this yields approximately normally distributed errors for each v_k

Computing confidence intervals

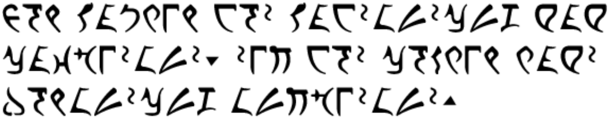
- The score v_k of any text represents a weighted mean
- LBG (2003) used this logic to develop a standard error of this mean using a *weighted variance* of the scores in the virgin text
- Given some assumptions about the scores being fixed (and the words being conditionally independent), this yields approximately normally distributed errors for each v_k
- An alternative would be to bootstrap the textual data prior to constructing C_{ij} and C_{kj} — see Lowe and Benoit (2012)

Pros and Cons of the Wordscores approach

- Fully automated technique with minimal human intervention or judgment calls – only with regard to reference text selection
- Language-blind: all we need to know are reference scores

Pros and Cons of the Wordscores approach

- Fully automated technique with minimal human intervention or judgment calls – only with regard to reference text selection
- Language-blind: all we need to know are reference scores
- Could potentially work on texts like this:



(See http://www.kli.org)

Pros and Cons of the Wordscores approach

- Estimates unknown positions on a priori scales – hence no inductive scaling with a posteriori interpretation of unknown policy space

Pros and Cons of the Wordscores approach

- Estimates unknown positions on a priori scales – hence no inductive scaling with a posteriori interpretation of unknown policy space
- Very dependent on correct identification of:
  - appropriate reference texts
  - appropriate reference scores