# Day 7: Supervised Text Scaling

Kenneth Benoit

Essex Summer School 2014

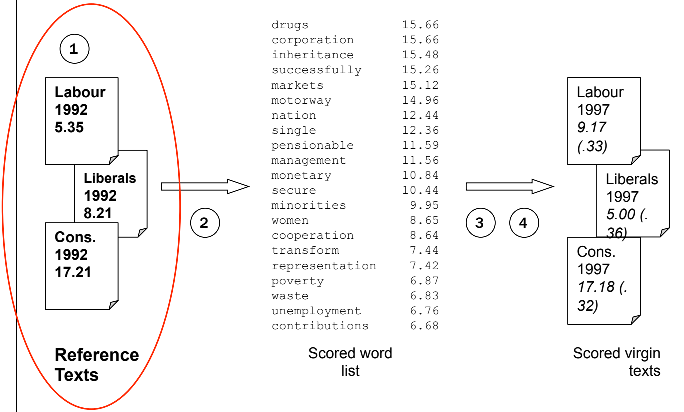
July 29, 2014

# 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

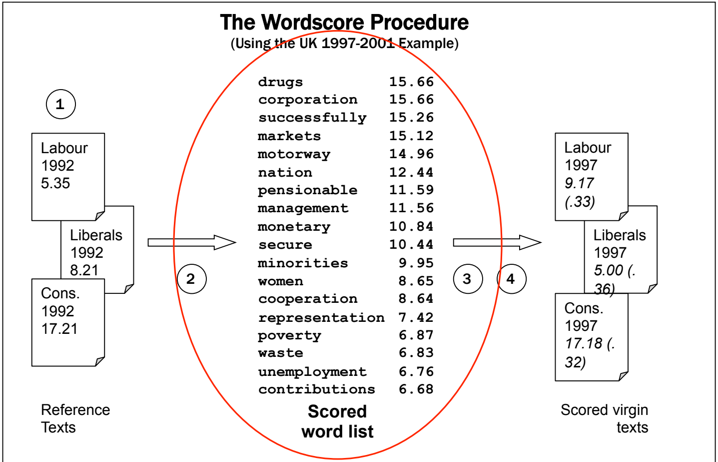
# Wordscores Procedure

**The Wordscore Procedure**
(Using the UK 1997-2001 Example)



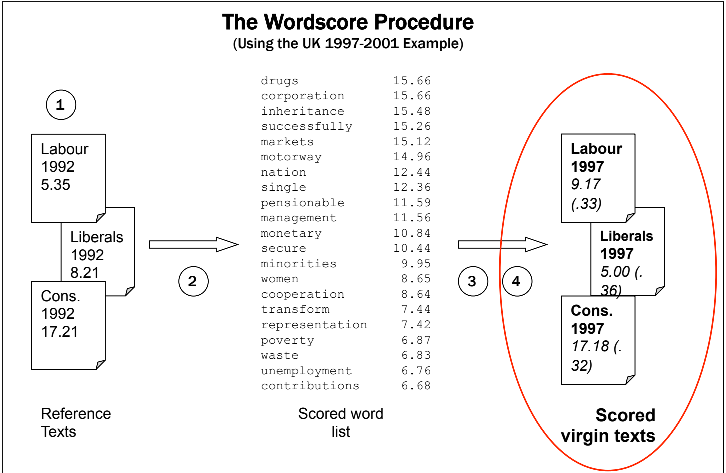

Step 1: Obtain reference texts with a priori known positions (`setref`)
Step 2: Generate word scores from reference texts (`wordscore`)
Step 3: Score each virgin text using word scores (`textscore`)
Step 4: (optional) Transform virgin text scores to original metric

# Wordscores Procedure



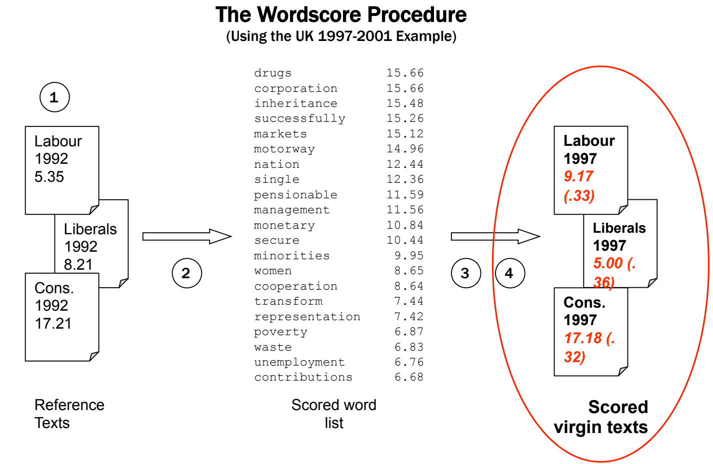

Step 1: Obtain reference texts with a priori known positions (`setref`)
Step 2: Generate word scores from reference texts (`wordscore`)
Step 3: Score each virgin text using word scores (`textscore`)
Step 4: (optional) Transform virgin text scores to original metric

# 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
- 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:

$$F_{ij} = \frac{C_{ij}}{C_{i\cdot}} \qquad (1)$$

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

$$P_{ij} = \frac{F_{ij}}{\sum_{i=1}^{I} F_{ij}} \tag{2}$$

- 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

$$P_{A\ \text{"choice"}} = \frac{.010}{(.010 + .030)} = 0.25 \tag{3}$$

$$P_{B\ \text{"choice"}} = \frac{.030}{(.010 + .030)} = 0.75 \tag{4}$$

# 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
- 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)
- Note also that nothing prohibits reference documents from also being scored as virgin documents

# Wordscores mathematically: Rescaling raw text scores

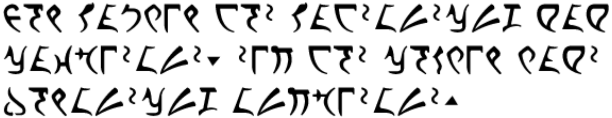
- Because of overlapping or non-discriminating words, the raw text scores will be dragged to the interior of the reference scores (we will see this shortly in the results)
- Some procedures can be applied to rescale them, either to a unit normal metric or to a more "natural" metric
- Martin and Vanberg (2008) have proposed alternatives to the LBG (2003) rescaling

# 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
- 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
- Very dependent on correct identification of:
  - appropriate reference texts
  - appropriate reference scores

# Suggestions for choosing reference texts

- Texts need to contain information representing a clearly dimensional position
- Dimension must be known a priori. Sources might include:
  - Survey scores or manifesto scores
  - Arbitrarily defined scales (e.g. -1.0 and 1.0)
- Should be as discriminating as possible: extreme texts on the dimension of interest, to provide reference anchors
- Need to be from the same lexical universe as virgin texts
- Should contain lots of words

# Suggestions for choosing reference values

- Must be "known" through some trusted external source
- For any pair of reference values, all scores are simply linear rescalings, so might as well use (-1, 1)
- The "middle point" will not be the midpoint, however, since this will depend on the relative word frequency of the reference documents
- Reference texts if scored as virgin texts will have document scores more extreme than other virgin texts
- With three or more reference values, the mid-point is mapped onto a multi-dimensional simplex. The values now matter but only in relative terms (we are still investigating this fully)

# Multinomial Bayes model of Class given a Word

## Class posterior probabilities

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

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

- 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
- 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:

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

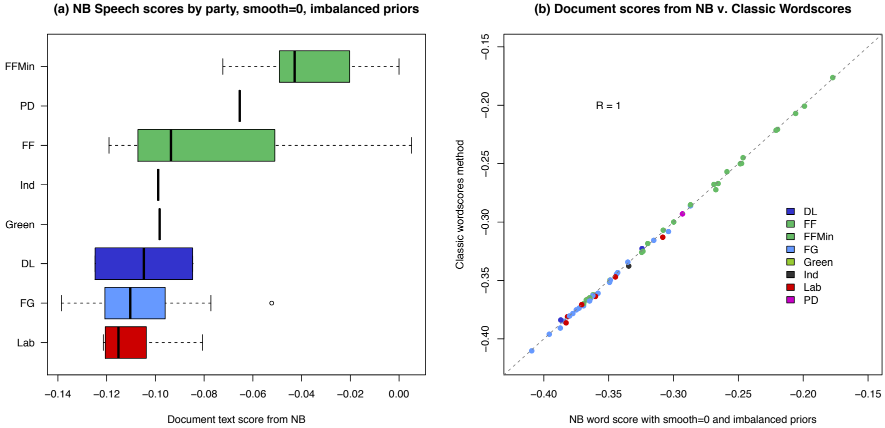
- So we *could* consider a document-level relative score, e.g. $1 - 2P(c_1|d)$ (for a two-class problem)
- But this turns out to be *useless*, since the predictions of class are highly separated

# 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)
- Beauchamp (2012) proposes a "Bayesscore" which is the arithmetic mean of the log difference word scores in a document – which yields extremely similar results

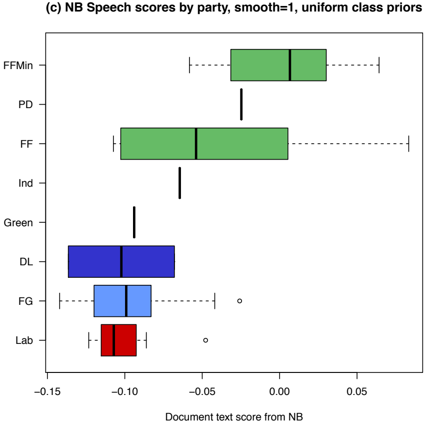
And now for some demonstrations with data...

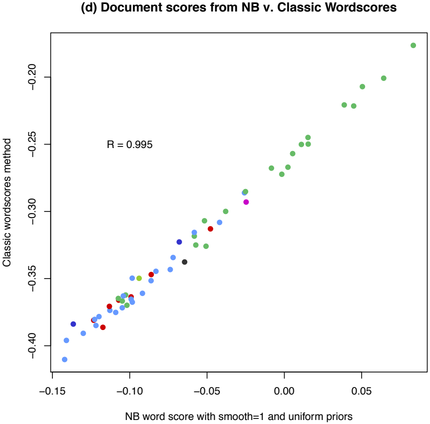
# Application 1: Dail speeches from LBG (2003)




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

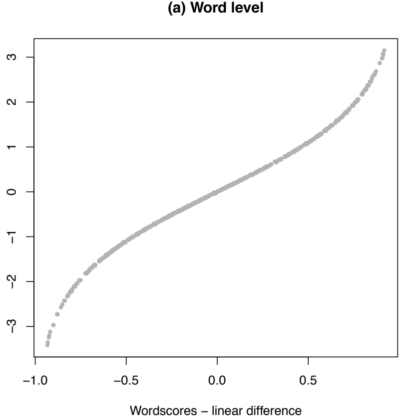
# Application 1: Dail speeches from LBG (2003)

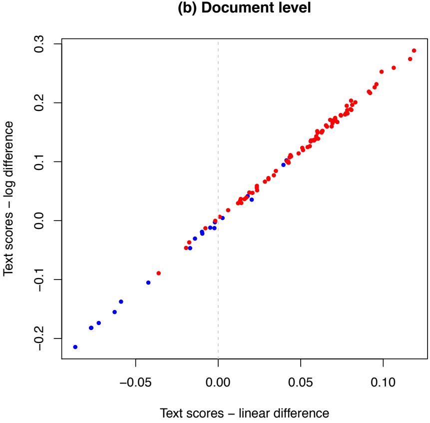



- two reference classes (Opposition+Opposition, Government) at {-1, 1}
- Laplace smoothing

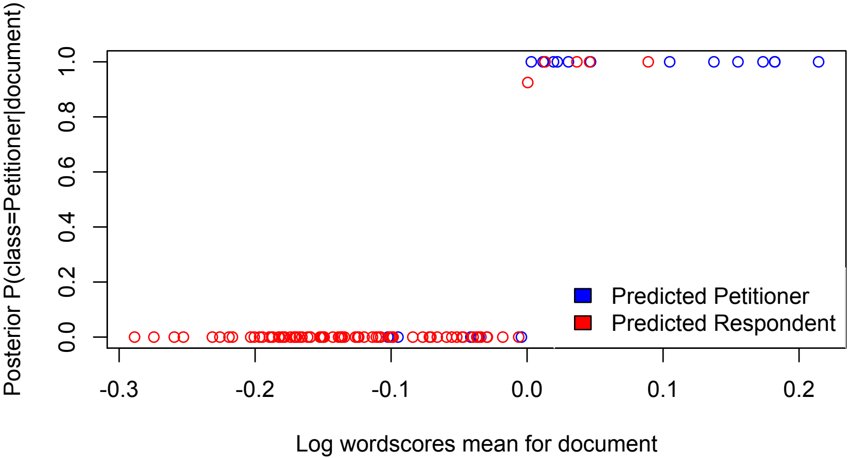
# Application 2: Classifying legal briefs (Evans et al 2007) Wordscores v. Bayesscore



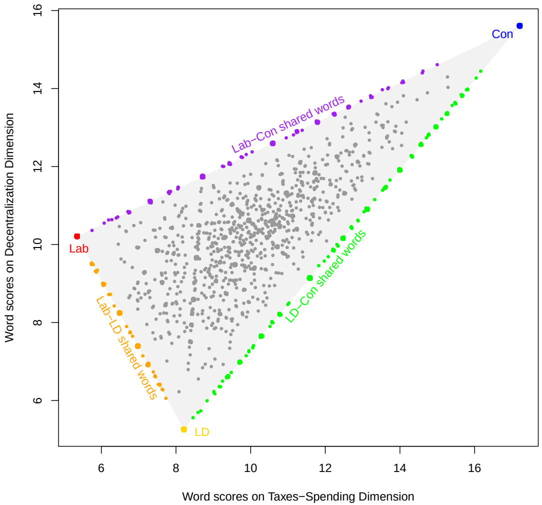




- Training set: Petitioner and Respondent litigant briefs from *Grutter/Gratz v. Bollinger* (a U.S. Supreme Court case)
- Test set: 98 amicus curiae briefs (whose P or R class is known)

# 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

| *Parameters* | | | | *Method* | | | |
|---|---|---|---|---|---|---|---|
| | | | | 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
- 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