Day 10: Text classification and scaling

ME314: Introduction to Data Science and Big Data Analytics

LSE Methods Summer Programme 2019

13 August 2019
Day 10 Outline

Overview of Supervised Learning vs.
Unsupervised Learning
Dictionary Methods
Scaling

Text Classification
Naive Bayes
Regularized Regression
Support Vector Machines

Scaling
Wordscores
Wordfish
Correspondence Analysis
Overview of Machine Learning Methods for Text Analysis
Supervised machine learning

**Goal:** classify documents into pre-existing categories.
e.g. authors of documents, sentiment of tweets, ideological position of parties based on manifestos, tone of movie reviews...

**What we need:**
- Hand-coded dataset (labeled), to be split into:
  - Training set: used to train the classifier
  - Validation/Test set: used to validate the classifier
- Method to extrapolate from hand coding to unlabeled documents (classifier):
  - Naive Bayes, regularized regression, SVM, K-nearest neighbors, BART, ensemble methods...
- Approach to validate classifier: cross-validation
- Performance metric to choose best classifier and avoid overfitting: confusion matrix, accuracy, precision, recall...
Creating a labeled set

How do we obtain a labeled set?

▶ **External sources of annotation**
  ▶ Disputed authorship of Federalist papers estimated based on known authors of other documents
  ▶ Party labels for election manifests
  ▶ Legislative proposals by think tanks (text reuse)

▶ **Expert annotation**
  ▶ “Canonical” dataset in Comparative Manifesto Project
  ▶ In most projects, undergraduate students (expertise comes from training)

▶ **Crowd-sourced coding**
  ▶ *Wisdom of crowds*: aggregated judgments of non-experts converge to judgments of experts at much lower cost (Benoit et al, 2016)
  ▶ Easy to implement with CrowdFlower or MTurk
Crowd-sourced text analysis (Benoit et al, 2016 APSR)

FIGURE 3. Expert and Crowd-sourced Estimates of Economic and Social Policy Positions

- Manifesto Placement Economic
- Manifesto Placement Social
FIGURE 5. Standard Errors of Manifesto-level Policy Estimates as a Function of the Number of Workers, for the Oversampled 1987 and 1997 Manifestos

Note: Each point is the bootstrapped standard deviation of the mean of means aggregate manifesto scores, computed from sentence-level random n subsamples from the codes.
Evaluating the quality of a labeled set

Any labeled set should be tested and reported for its inter-rate reliability, at three different standards:

<table>
<thead>
<tr>
<th>Type</th>
<th>Test Design</th>
<th>Causes of Disagreements</th>
<th>Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>test-retest</td>
<td>intraobserver inconsistencies</td>
<td>weakest</td>
</tr>
<tr>
<td>Reproducibility</td>
<td>test-test</td>
<td>intraobserver inconsistencies + interobserver disagreements</td>
<td>medium</td>
</tr>
<tr>
<td>Accuracy</td>
<td>test-standard</td>
<td>intraobserver inconsistencies + interobserver disagreements + deviations from a standard</td>
<td>strongest</td>
</tr>
</tbody>
</table>
Measures of agreement

- **Percent agreement** Very simple:
  \[
  \frac{\text{number of agreeing ratings}}{\text{total ratings}} \times 100\%
  \]

- **Correlation**
  - (usually) Pearson’s $r$, aka product-moment correlation
  - Formula: \[
  r_{AB} = \frac{1}{n-1} \sum_{i=1}^{n} \left( \frac{A_i - \bar{A}}{s_A} \right) \left( \frac{B_i - \bar{B}}{s_B} \right)
  \]
  - May also be ordinal, such as Spearman’s rho or Kendall’s tau-b
  - Range is [0,1]

- **Agreement measures**
  - Take into account not only observed agreement, but also agreement that would have occurred by chance
  - Cohen’s $\kappa$ is most common
  - Krippendorff’s $\alpha$ is a generalization of Cohen’s $\kappa$
  - Both range from [0,1]
Reliability data matrixes

Example here used binary data (from Krippendorff)

<table>
<thead>
<tr>
<th>Article:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coder A</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Coder B</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- A and B agree on 60% of the articles: 60% agreement
- Correlation is (approximately) 0.10
- Observed disagreement: 4
- Expected disagreement (by chance): 4.4211
- Krippendorff’s $\alpha = 1 - \frac{D_o}{D_e} = 1 - \frac{4}{4.4211} = 0.095$
- Cohen’s $\kappa$ (nearly) identical
Basic principles of supervised learning

- **Generalization**: A classifier or a regression algorithm learns to correctly predict output from given inputs not only in previously seen samples but also in previously unseen samples.

- **Overfitting**: A classifier or a regression algorithm learns to correctly predict output from given inputs in previously seen samples but fails to do so in previously unseen samples. This causes poor prediction/generalization.

- Goal is to maximize the frontier of precise identification of true condition with accurate recall.
Measuring performance

- Classifier is trained to maximize in-sample performance
- But generally we want to apply method to new data
- Danger: overfitting

- Model is too complex, describes noise rather than signal (Bias-Variance trade-off)
- Focus on features that perform well in labeled data but may not generalize (e.g. “inflation” in 1980s)
- In-sample performance better than out-of-sample performance

- Solutions?
  - Randomly split dataset into training and test set
  - Cross-validation
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 exemplify contrasting *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, regularized regression, Support Vector Machines (SVM)
  ▶ **Unsupervised**: topic models, IRT models, correspondence analysis, factor analytic approaches

▶ Political science applications
  ▶ **Supervised**: Wordscores (LBG 2003); SVMs (Yu, Kaufman and Diermeier 2008); Naive Bayes (Evans et al 2007)
  ▶ **Unsupervised**: Structural topic model (Roberts et al 2014); “Wordfish” (Slapin and Proksch 2008); two-dimensional IRT (Monroe and Maeda 2004)
Supervised learning v. dictionary methods

- Dictionary methods:
  - Advantage: not corpus-specific, cost to apply to a new corpus is trivial
  - Disadvantage: not corpus-specific, so performance on a new corpus is unknown (domain shift)

- Supervised learning can be conceptualized as a generalization of dictionary methods, where features associated with each categories (and their relative weight) are learned from the data

- By construction, they will outperform dictionary methods in classification tasks, as long as training sample is large enough
Dictionaries vs supervised learning

Lexicons’ Accuracy in Document Classification Compared to Machine-Learning Approach

Source: González-Bailón and Paltoglou (2015)
### Dictionaries vs supervised learning

**Application: sentiment analysis of NYTimes articles**

<table>
<thead>
<tr>
<th>Method</th>
<th>Undergraduates</th>
<th>Crowd</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML (crowd-sourcing)</td>
<td>74.0</td>
<td>71.0</td>
</tr>
<tr>
<td>ML (with directional terms)</td>
<td>74.4</td>
<td>70.5</td>
</tr>
<tr>
<td>Dictionary (SentiStrength)</td>
<td>57.2</td>
<td>56.4</td>
</tr>
<tr>
<td>Dictionary (LexiCoder)</td>
<td>57.6</td>
<td>56.6</td>
</tr>
<tr>
<td>9-Word Method (Hopkins, 2010)</td>
<td>35.6</td>
<td>39.9</td>
</tr>
</tbody>
</table>

Accuracy (Proportion of Articles Correctly Predicted)

*Source: Barberá et al (2017)*
Dictionaries vs supervised learning

Application: sentiment analysis of NYTimes articles

- ML (crowd-sourcing)
- ML (with directional terms)
- Dictionary (SentiStrength)
- Dictionary (LexiCoder)
- 9-Word Method (Hopkins, 2010)

Source: Barberá et al (2017)
Classification v. scaling methods compared

- Machine learning focuses on identifying classes (classification), while social science is typically interested in locating things on latent traits (scaling).
- But the two methods overlap and can be adapted – will demonstrate later using the Naive Bayes classifier.
- Applying lessons from machine learning to supervised scaling, we can
  - Apply classification methods to scaling
  - Improve it using lessons from machine learning
Text Classification
Types of classifiers

General thoughts:
- Trade-off between accuracy and interpretability
- Parameters need to be cross-validated

Frequently used classifiers:
- Naive Bayes
- Regularized regression
- SVM
- Others: k-nearest neighbors, tree-based methods, etc.
- Ensemble methods
Consider $J$ word types distributed across $N$ documents, each assigned one of $K$ classes.

At the word level, Bayes Theorem tells us that:

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

For two classes, this can be expressed as

$$= \frac{P(w_j|c_k)P(c_k)}{P(w_j|c_k)P(c_k) + P(w_j|c_{\neg k})P(c_{\neg k})} \quad (1)$$
Multinomial Bayes model of Class given a Word
Class-conditional word likelihoods

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

- The word likelihood within class
- The maximum likelihood estimate is simply the proportion of times that word \( j \) occurs in class \( k \), but it is more common to use Laplace smoothing by adding 1 to each observed count within class
Multinomial Bayes model of Class given a Word

Word probabilities

\[ P(c_k | w_j) = \frac{P(w_j | c_k)P(c_k)}{P(w_j)} \]

- This represents the word probability from the training corpus
- Usually uninteresting, since it is constant for the training data, but needed to compute posteriors on a probability scale
Multinomial Bayes model of Class given a Word

Class prior probabilities

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

- This represents the class prior probability
- Machine learning typically takes this as the document frequency in the training set
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 \( j \)

- Key for the classifier: in new documents, we only observe word distributions and want to predict class
Moving to the document level

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

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

\[ P(c|d) \propto P(c) \prod_j P(w_j|c) \]

- This is why we call it “naive”: because it (wrongly) assumes:
  - conditional independence of word counts
  - positional independence of word counts
Naive Bayes Classification Example

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

### Table 13.1 Data for parameter estimation examples.

<table>
<thead>
<tr>
<th>docID</th>
<th>words in document</th>
<th>in $c = \text{China}$?</th>
</tr>
</thead>
<tbody>
<tr>
<td>training set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Chinese Beijing Chinese</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Chinese Chinese Shanghai</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>Chinese Macao</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>Tokyo Japan Chinese</td>
<td>no</td>
</tr>
<tr>
<td>test set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Chinese Chinese Chinese Tokyo Japan</td>
<td>?</td>
</tr>
</tbody>
</table>
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:

\[
\begin{align*}
\hat{P}(\text{Chinese}|c) &= \frac{5 + 1}{8 + 6} = \frac{6}{14} = \frac{3}{7} \\
\hat{P}(\text{Tokyo}|c) &= \hat{P}(\text{Japan}|c) = \frac{0 + 1}{8 + 6} = \frac{1}{14} \\
\hat{P}(\text{Chinese}|\overline{c}) &= \frac{1 + 1}{3 + 6} = \frac{2}{9} \\
\hat{P}(\text{Tokyo}|\overline{c}) &= \hat{P}(\text{Japan}|\overline{c}) = \frac{1 + 1}{3 + 6} = \frac{2}{9}
\end{align*}
\]

The denominators are $(8 + 6)$ and $(3 + 6)$ because the lengths of $\text{text}_c$ and $\text{text}_{\overline{c}}$ 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:

\[
\begin{align*}
\hat{P}(c|d_5) &\propto \frac{3}{4} \cdot \left(\frac{3}{7}\right)^3 \cdot \frac{1}{14} \cdot \frac{1}{14} \approx 0.0003. \\
\hat{P}(\overline{c}|d_5) &\propto \frac{1}{4} \cdot \left(\frac{2}{9}\right)^3 \cdot \frac{2}{9} \cdot \frac{2}{9} \approx 0.0001.
\end{align*}
\]

Thus, the classifier assigns the test document to $c = \text{China}$. 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.
Regularized regression

Assume we have:

- \( i = 1, 2, \ldots, N \) documents
- Each document \( i \) is in class \( y_i = 0 \) or \( y_i = 1 \)
- \( j = 1, 2, \ldots, J \) unique features
- And \( x_{ij} \) as the count of feature \( j \) in document \( i \)

We could build a linear regression model as a classifier, using the values of \( \beta_0, \beta_1, \ldots, \beta_J \) that minimize:

\[
RSS = \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2
\]

But can we?

- If \( J > N \), OLS does not have a unique solution
- Even with \( N > J \), OLS has low bias/high variance (overfitting)
Regularized regression

What can we do? Add a **penalty for model complexity**, such that we now minimize:

\[
N \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \beta_j^2 \rightarrow \text{ridge regression}
\]

or

\[
N \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} |\beta_j| \rightarrow \text{lasso regression}
\]

where \( \lambda \) is the **penalty parameter** (to be estimated)
Regularized regression

Why the penalty (shrinkage)?

▶ Reduces the variance
▶ Identifies the model if $J > N$
▶ Some coefficients become zero (feature selection)

The penalty can take different forms:

▶ **Ridge regression**: $\lambda \sum_{j=1}^{J} \beta_j^2$ with $\lambda > 0$; and when $\lambda = 0$ becomes OLS
▶ **Lasso** $\lambda \sum_{j=1}^{J} |\beta_j|$ where some coefficients become zero.
▶ **Elastic Net**: $\lambda_1 \sum_{j=1}^{J} \beta_j^2 + \lambda_2 \sum_{j=1}^{J} |\beta_j|$ (best of both worlds?)

How to find best value of $\lambda$? Cross-validation.

**Evaluation**: regularized regression is easy to interpret, but often outperformed by more complex methods.
Intuition: finding classification boundary that best separates observations of different classes.

Harder to visualize in more than two dimensions (hyperplanes)
Support Vector Machines

With no perfect separation, goal is to minimize distances to marginal points, conditioning on a tuning parameter $C$ that indicates tolerance to errors (controls bias-variance trade-off)
In previous examples, vectors were linear; but we can try different kernels (polynomial, radial):

And of course we can have multiple vectors within same classifier.
Ensemble methods

Intuition:

- Fit multiple classifiers, different types
- Test how well they perform in test set
- For new observations, produce prediction aggregating predictions of individual classifiers
- How to aggregate predictions?
  - Pick best classifier
  - Average of predicted probabilities
  - Weighted average (weights proportional to classification error)
- Implement in SuperLearner package in R
Scaling
Supervised scaling methods

Wordscores method (Laver, Benoit & Garry, 2003):

- 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 we’d 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

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

1. Labour 1992 5.35
2. Liberals 1992 8.21
3. Cons. 1992 17.21
4. Labour 1997 9.17 (.33)
5. Liberals 1997 5.00 (.36)
6. Cons. 1997 17.18 (.32)

**Scored word list**
- drugs 15.66
- corporation 15.66
- successfully 15.26
- markets 15.12
- motorway 14.96
- nation 12.44
- pensionable 11.59
- management 11.56
- monetary 10.84
- secure 10.44
- minorities 9.95
- women 8.65
- cooperation 8.64
- representation 7.42
- poverty 6.87
- waste 6.83
- unemployment 6.76
- contributions 6.68

**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

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

1. Reference Texts

2. Scored word list

3. Scored virgin texts

4. Labour 1997 9.17 (.33)

   Liberals 1997 5.00 (.36)

   Cons. 1997 17.18 (.32)

Drugs 15.66
Corporation 15.66
Inheritance 15.48
Successfully 15.26
Markets 15.12
Motorway 14.96
Nation 12.44
Single 12.36
Pensionable 11.59
Management 11.56
Monetary 10.84
Secure 10.44
Minorities 9.95
Women 8.65
Cooperation 8.64
Transform 7.44
Representation 7.42
Poverty 6.87
Waste 6.83
Unemployment 6.76
Contributions 6.68

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
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-feature 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-feature matrix within each document by converting $C_{ij}$ into a relative document-feature matrix (within document), by dividing $C_{ij}$ by its word total marginals:

  $$F_{ij} = \frac{C_{ij}}{C_i}.$$  \hspace{1cm} (2)

  where $C_i = \sum_{j=1}^{J} C_{ij}$.
Wordscores mathematically: Word scores

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

\[
P_{ij} = \frac{F_{ij}}{\sum_{i=1}^{I} F_{ij}}
\]

(3)

> 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 $i$
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

\[
P_A \text{"choice"} = \frac{.010}{(.010 + .030)} = 0.25
\]

\[
P_B \text{"choice"} = \frac{.030}{(.010 + .030)} = 0.75
\]
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 $P_{ij}$:

\[
S_j = \sum_{i=1}^{l} a_i P_{ij}
\]  

(6)

- In matrix algebra, $S = a \cdot P$

- 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
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:

$$ v_k = \sum_j (F_{kj} \cdot s_j) $$  \hspace{1cm} (7)

where $F_{kj} = \frac{C_{kj}}{C_k}$ 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

- 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 \( P(c_{\neg k}|w_j) \)
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

$$s_j^* = s_1 P_1 + s_2 P_2$$
$$= s_1 P_1 + s_2 (1 - P_1)$$
$$= s_1 P_1 + s_2 - s_2 P_1$$
$$= P_1 (s_1 - s_2) + s_2$$
“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

\[
 s_j^* = s_1 P_1 + s_2 P_2 + \cdots + s_K P_K \\
= s_1 P_1 + s_2 P_2 + \cdots + s_K \left( 1 - \sum_{k=1}^{K-1} P_k \right) \\
= \sum_{k=1}^{K-1} P_i (s_k - s_K) + s_I
\]
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 $s_1 = -1.0, s_2 = 1.0$,

$$s_j^B = -\log P_1 + \log P_2$$

$$= \log \frac{1 - P_1}{P_1}$$
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)

(a) NB Speech scores by party, smooth=0, imbalanced priors

(b) Document scores from NB v. Classic Wordscores

- three reference classes (Opposition, Opposition, Government) at \{-1, -1, 1\}
- no smoothing
Application 1: Daily 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

(a) Word level

(b) Document level

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

![Graph showing posterior class prediction from NB versus log wordscores. The x-axis represents the log wordscores mean for the document, and the y-axis represents the posterior probability. The graph compares predicted Petitioner and predicted Respondent classes using blue and red markers, respectively.]
Application 3: Scaling environmental interest groups (Klüver 2009)

- Dataset: text of online consultation on EU environmental regulations
- Reference texts: most extreme pro- and anti-regulation groups
Application 4: 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\}
Unsupervised methods scale distance

- Text gets converted into a quantitative matrix of features
  - words, typically
  - could be dictionary entries, or parts of speech
- Documents are scaled based on similarity or distance in feature use
- Fundamental problem: distance on which scale?
  - Ideally, something we care about, e.g. policy positions, ideology, preferences, sentiment
  - But often other dimensions (language, rhetoric style, authorship) are more predictive
- First dimension in unsupervised scaling will capture main source of variation, whatever that is
- Unlike supervised models, validation comes after estimating the model
Unsupervised scaling methods

Two main approaches

- **Parametric methods** model feature occurrence according to some stochastic distribution, typically in the form of a measurement model
  - for instance, model words as a multi-level Bernoulli distribution, or a Poisson distribution
  - word effects and “positional” effects are unobserved parameters to be estimated
  - e.g. Wordfish (Slapin and Proksch 2008) and Wordshoal (Lauderdale and Herzog 2016)

- **Non-parametric methods** typically based on the Singular Value Decomposition of a matrix
  - correspondence analysis
  - factor analysis
  - other (multi)dimensional scaling methods
Wordfish (Slapin and Proksch 2008)

- Goal: unsupervised scaling of ideological positions
- The frequency with which politician \( i \) uses word \( k \) is drawn from a Poisson distribution:

\[
 w_{ik} \sim \text{Poisson}(\lambda_{ik})
\]

\[
 \lambda_{ik} = \exp(\alpha_i + \psi_k + \beta_k \times \theta_i)
\]

- with latent parameters:
  - \( \alpha_i \) is “loquaciousness” of politician \( i \)
  - \( \psi_k \) is frequency of word \( k \)
  - \( \beta_k \) is discrimination parameter of word \( k \)
  - \( \theta_i \) is the politician’s ideological position
- Key intuition: controlling for document length and word frequency, words with negative \( \beta_k \) will tend to be used more often by politicians with negative \( \theta_i \) (and vice versa)
Why Poisson?

- Poisson-distributed variables are bounded between \((0, \infty)\) and take on only discrete values \(0, 1, 2, \ldots, \infty\)

- Exponential transformation: word counts are function of log document length and word frequency

\[
\begin{align*}
  w_{ik} &\sim \text{Poisson}(\lambda_{ik}) \\
  \lambda_{ik} &= \exp(\alpha_i + \psi_k + \beta_k \times \theta_i) \\
  \log(\lambda_{ik}) &= \alpha_i + \psi_k + \beta_k \times \theta_i
\end{align*}
\]
How to estimate this model

Conditional maximum likelihood estimation:

- If we knew $\psi$ and $\beta$ (the word parameters) then we have a Poisson regression model.
- If we knew $\alpha$ and $\theta$ (the party / politician / document parameters) then we have a Poisson regression model too!
- So we alternate them and hope to converge to reasonable estimates for both.
- Implemented in the quanteda package as `textmodel_wordfish`.

An alternative is MCMC with a Bayesian formulation or variational inference using an Expectation-Maximization algorithm (Imai et al 2016).
Start by guessing the parameters (some guesses are better than others, e.g. SVD)

Algorithm:

1. Assume the current legislator parameters are correct and fit as a Poisson regression model
2. Assume the current word parameters are correct and fit as a Poisson regression model
3. Normalize $\theta$s to mean 0 and variance 1

Iterate until convergence (change in values is below a certain threshold)
Identification

The *scale* and *direction* of $\theta$ is undetermined — like most models with latent variables

To identify the model in Wordfish

- Fix one $\alpha$ to zero to specify the left-right direction (Wordfish option 1)
- Fix the $\hat{\theta}$s to mean 0 and variance 1 to specify the scale (Wordfish option 2)
- Fix two $\hat{\theta}$s to specify the direction and scale (Wordfish option 3 and Wordscores)

Note: Fixing two reference scores does not specify the policy domain, it just identifies the model
“Features” of the parametric scaling approach

- Standard (statistical) inference about parameters
- Uncertainty accounting for parameters
- Distributional assumptions are made explicit (as part of the data generating process motivating the choice of stochastic distribution)
  - conditional independence
  - stochastic process (e.g. $E(Y_{ij}) = \text{Var}(Y_{ij}) = \lambda_{ij}$)
- Permits hierarchical reparameterization (to add covariates)
- Generative model: given the estimated parameters, we could generate a document for any specified length
Some reasons why this model is wrong

- Violations of conditional independence:
  - Words occur in sequence (serial correlation)
  - Words occur in combinations (e.g. as collocations)
    “carbon tax” / “income tax” / “inheritance tax” / “capital gains tax” / “bank tax”
  - Legislative speech uses rhetoric that contains frequent synonyms and repetition for emphasis (e.g. “Yes we can!”)

- Heteroskedastic errors (variance not constant and equal to mean):
  - **over**dispersion when “informative” words tend to cluster together
  - **under**dispersion could (possibly) occur when words of high frequency are uninformative and have relatively low between-text variation (once length is considered)
Overdispersion in German manifesto data
(data taken from Slapin and Proksch 2008)
One solution to model overdispersion

Negative binomial model (Lo, Proksch, and Slapin 2014):

\[ w_{ik} \sim \text{NB} \left( r, \frac{\lambda_{ik}}{\lambda_{ik} + r_i} \right) \]

\[ \lambda_{ik} = \exp(\alpha_i + \psi_k + \beta_k \times \theta_i) \]

where \( r_i \) is a variance inflation parameter that varies across documents.

It can have a substantive interpretation (ideological ambiguity), e.g. when a party emphasizes an issue but fails to mention key words associated with it that a party with similar ideology mentions.
Example from Slapin and Proksch 2008

Figure 1: Estimated Party Positions in Germany, 1990–2005

(A) Left–Right

(B) Economic Policy

(C) Societal Policy

(D) Foreign Policy
Figure 2  Word Weights vs. Word Fixed Effects. Left-Right Dimension, Germany 1990–2005 (Translations given in text)
### Table 1  Top 10 Words Placing Parties on the Left and Right

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Left-Right</strong></td>
<td>Federal Republic of Germany (BRD)</td>
<td>general welfare payments (Bürgergeldsystem)</td>
</tr>
<tr>
<td></td>
<td>immediate (sofortiger)</td>
<td>introduction (Heranführung)</td>
</tr>
<tr>
<td></td>
<td>pornography (Pornographie)</td>
<td>income taxation (Einkommensbesteuerung)</td>
</tr>
<tr>
<td></td>
<td>sexuality (Sexualität)</td>
<td>non-wage labor costs (Lohnzusatzkosten)</td>
</tr>
<tr>
<td></td>
<td>substitute materials (Ersatzstoffen)</td>
<td>business location (Wirtschaftsstandort)</td>
</tr>
<tr>
<td></td>
<td>stratosphere (Stratosphäre)</td>
<td>university of applied sciences (Fachhochschule)</td>
</tr>
<tr>
<td></td>
<td>women’s movement (Frauenbewegung)</td>
<td>education vouchers (Bildungsgutscheine)</td>
</tr>
<tr>
<td></td>
<td>fascism (Faschismus)</td>
<td>mobility (Beweglichkeit)</td>
</tr>
<tr>
<td></td>
<td>Two thirds world (Zweidrittelwelt)</td>
<td>peace tasks (Friedensaufgaben)</td>
</tr>
<tr>
<td></td>
<td>established (etablierten)</td>
<td>protection (Protektion)</td>
</tr>
<tr>
<td><strong>Economic</strong></td>
<td>Federal Republic of Germany (BRD)</td>
<td>to seek (anzustreben)</td>
</tr>
<tr>
<td></td>
<td>democratization (Demokratisierung)</td>
<td>general welfare payments (Bürgergeldsystem)</td>
</tr>
<tr>
<td></td>
<td>to prohibit (verbieten)</td>
<td>inventors (Erfinder)</td>
</tr>
<tr>
<td></td>
<td>destruction (Zerstörung)</td>
<td>mobility (Beweglichkeit)</td>
</tr>
<tr>
<td></td>
<td>mothers (Mütter)</td>
<td>location (Standorts)</td>
</tr>
<tr>
<td></td>
<td>debasing (entwürdigende)</td>
<td>negotiated wages (Tarif-Löhne)</td>
</tr>
<tr>
<td></td>
<td>weeks (Wochen)</td>
<td>child-raising allowance (Erziehungsgeld)</td>
</tr>
<tr>
<td></td>
<td>quota (Quotierung)</td>
<td>utilization (Verwertung)</td>
</tr>
<tr>
<td></td>
<td>unprotected (ungeschützter)</td>
<td>savings (Ersparnis)</td>
</tr>
<tr>
<td></td>
<td>workers’ participation (Mitbestimmungsmöglichkeiten)</td>
<td>reliable (verlässiglich)</td>
</tr>
</tbody>
</table>
### Table 2  Cross-Validation: Correlations between German Party Position Estimates

<table>
<thead>
<tr>
<th></th>
<th>Poisson Scaling Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left-Right</td>
</tr>
<tr>
<td>Hand-coding manifestos</td>
<td></td>
</tr>
<tr>
<td>CMP: Left-Right (n = 15, 1990–1998)</td>
<td>−0.82</td>
</tr>
<tr>
<td>CMP: Markeco (n = 15, 1990–1998)</td>
<td>0.81</td>
</tr>
<tr>
<td>CMP: Welfare (n = 15, 1990–1998)</td>
<td>0.58</td>
</tr>
<tr>
<td>CMP: Intpeace (n = 15, 1990–1998)</td>
<td>0.81</td>
</tr>
<tr>
<td>Expert Survey</td>
<td></td>
</tr>
<tr>
<td>Benoit/Laver 2006: Left-Right (n = 5, 2002)</td>
<td>−0.91</td>
</tr>
<tr>
<td>Benoit/Laver 2006: Taxes-Spending (n = 5, 2002)</td>
<td>0.86</td>
</tr>
<tr>
<td>Wordscores</td>
<td></td>
</tr>
<tr>
<td>Laver et al. 2003: Economic (n = 10, 1990–1994)</td>
<td>0.93</td>
</tr>
<tr>
<td>Laver et al. 2003: Social (n = 10, 1990–1994)</td>
<td></td>
</tr>
<tr>
<td>Proksch/Slapin 2006: Economic (n = 5, 2005)</td>
<td>0.98</td>
</tr>
<tr>
<td>Proksch/Slapin 2006: Social (n = 5, 2005)</td>
<td>−0.47</td>
</tr>
</tbody>
</table>
Two key **limitations** of wordfish applied to legislative text:

- Word discrimination parameters assumed to be constant across debates (unrealistic, think e.g. “debt”)
- May not capture left-right ideology but topic variation

Slapin and Proksch partially avoid these issues by scaling different types of debates separately.

But resulting estimates are confined to set of speakers who spoke on each topic.

Wordshoal solution: aggregate debate-specific ideal points into a reduced number of scales.
The frequency with which politician $i$ uses word $k$ in debate $j$ is drawn from a Poisson distribution:

$$w_{ijk} \sim \text{Poisson}(\lambda_{ijk})$$

$$\lambda_{ijk} = \exp(\alpha_{ij} + \psi_{jk} + \beta_{kj} \times \theta_{ij})$$

$$\theta_{ij} \sim \mathcal{N}(\nu_j + \kappa_j \mu_i, \tau_i)$$

with latent parameters:
- $\alpha_{ij}$ is “loquaciousness” of politician $i$ in debate $j$
- $\psi_{jk}$ is frequency of word $k$ in debate $j$
- $\beta_{kj}$ is discrimination parameter of word $k$ in debate $j$
- $\theta_{ij}$ is the politician’s ideological position in debate $j$
- $\nu_j$ is baseline ideological position of debate $j$
- $\kappa_j$ is correlation of debate $j$ with common dimension
- $\mu_i$ is overall ideological position of politician $i$

Intuition: debate-specific estimates are aggregated into a single position using dimensionality reduction
New quantities of interest to estimate:

- Politicians’ overall position vs debate-specific positions
- Strength of association between debate scales and general ideological scale
- Association of words with general scales, and stability of word discrimination parameters across debates
Example from Lauderdale and Herzog 2016
Example from Lauderdale and Herzog 2016

**Wordfish**

In

Cor = 0.31

Coalition Status

Out

-2 -1 0 1 2

Score

30th Dail

**Wordshoal**

In

Cor = 0.94

Coalition Status

Out

-1.0 0.0 0.5 1.0 1.5 2.0

Score

30th Dail

**Wordfish**

Legislator

Government

Opposition

Estimated Position

30th Dail

-3 -2 -1 0 1 2 3

**Wordshoal**

Legislator

Government

Opposition

Estimated Position

30th Dail

-4 -2 0 2 4
Example from Lauderdale and Herzog 2016
Example from Lauderdale and Herzog 2016

Table 2: The five debates with the highest and lowest loadings on the government versus opposition dimension, as measured by the absolute value of $\beta_j$ ranging from 0 to 1.

<table>
<thead>
<tr>
<th>High government-opposition polarization</th>
<th>Abs. $\beta_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social Welfare and Pensions (No. 2) Bill 2009 (Second Stage)</td>
<td>0.942</td>
</tr>
<tr>
<td>Early Childhood Care and Education (Motion)</td>
<td>0.887</td>
</tr>
<tr>
<td>Private Members’ Business – Vaccination Programme (Motion)</td>
<td>0.824</td>
</tr>
<tr>
<td>Capitation Grants (Motion)</td>
<td>0.819</td>
</tr>
<tr>
<td>Confidence in Government (Motion)</td>
<td>0.814</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Low government-opposition polarization</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancer Services Reports (Motion)</td>
<td>0.003</td>
</tr>
<tr>
<td>Finance (No. 2) Bill 2007 (Committee and Remaining Stages)</td>
<td>0.002</td>
</tr>
<tr>
<td>Finance Bill 2011 (Report and Final Stages)</td>
<td>0.002</td>
</tr>
<tr>
<td>Private Members’ Business – Mortgage Arrears (Motion)</td>
<td>0.002</td>
</tr>
<tr>
<td>Wildlife (Amendment) Bill 2010 (Committee and Remaining Stages)</td>
<td>0.001</td>
</tr>
</tbody>
</table>
Example from Lauderdale and Herzog 2016

Word Alignments over Time

Frequency-Weighted Average Word Loading

Clinton 104, Clinton 105, Clinton 106, Bush 107, Bush 108, Bush 109, Bush 110, Obama 111, Obama 112, Obama 113

Year


Words: deficit, health, preexist, tax, constituent
Non-parametric methods

- Non-parametric methods are algorithmic, involving no “parameters” in the procedure that are estimated.
- Hence there is no uncertainty accounting given distributional theory.
- Advantage: don’t have to make assumptions.
- Disadvantages:
  - cannot leverage probability conclusions given distributional assumptions and statistical theory.
  - results highly fit to the data.
  - not really assumption-free, if we are honest.
Correspondence Analysis

- CA is like factor analysis for categorical data
- Following normalization of the marginals, it uses Singular Value Decomposition to reduce the dimensionality of the document-feature matrix
- This allows projection of the positioning of the words as well as the texts into multi-dimensional space
- The number of dimensions – as in factor analysis – can be decided based on the eigenvalues from the SVD
Singular Value Decomposition

- A matrix $X_{n \times k}$ can be represented in a dimensionality equal to its rank $d$ as:

\[
X_{n \times k} = U_{n \times d} \Sigma_{d \times d} V'_{d \times k}
\]  

(1)

- The $U$, $\Sigma$, and $V$ matrixes “relocate” the elements of $X$ onto new coordinate vectors in $d$-dimensional Euclidean space.

- Row variables of $X$ become points on the $U$ column coordinates, and the column variables of $X$ become points on the $V$ column coordinates.

- The coordinate vectors are perpendicular (orthogonal) to each other and are normalized to unit length.
Correspondence analysis

1. Compute matrix of standardized residuals, $S$:
   \[ S = D_r^{1/2}(P - rc^T)D_c^{1/2} \]
   where
   \[ P = Y / \sum_{ij} y_{ij} \]
   $r, c$ are row/column masses: e.g. $r_i = \sum_j p_{ij}$
   $D_r = \text{diag}(r), D_c = \text{diag}(c)$

2. Calculate SVD of $S$

3. Project rows and columns onto low-dimensional space:
   \[ \theta = D_r^{1/2}U \text{ for rows (documents)} \]
   \[ \phi = D_c^{1/2}V \text{ for columns (words)} \]

Mathematically close to log-linear poisson regression model
(Lowe, 2008)
SCHONHARDT - BAILEY appears to be unique to that bill – i.e., the specific procedural measures, the constitutionality of the absent health exception, and the gruesome medical details of the procedure are all unique to the PBA ban as defined in the 2003 bill. Hence, to ignore the content of the debates by focusing solely on the final roll-call vote is to miss much of what concerned senators about this particular bill. To see this more clearly, we turn to Figure 3, in which the results from ALCESTE’s classification are represented in correspondence space.

Fig. 3. Correspondence analysis of classes and tags from Senate debates on Partial-Birth Abortion Ban Act

<table>
<thead>
<tr>
<th></th>
<th>Eigenvalue</th>
<th>% Association</th>
<th>% Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor 1</td>
<td>0.30</td>
<td>44.4</td>
<td>44.4</td>
</tr>
<tr>
<td>Factor 2</td>
<td>0.22</td>
<td>32.9</td>
<td>77.3</td>
</tr>
</tbody>
</table>

*Fig. 3. Correspondence analysis of classes and tags from Senate debates on Partial-Birth Abortion Ban Act*
Fig. 4. Senate debates on Partial-Birth Abortion Ban Act – word distribution in correspondence space