Text Analysis in R

Lecture 2: Analysis of text data

Helge Liebert

Outline

Representing text as data

Representing text as data

Introduction

Introduction

- Text differs from other, traditional forms of data.
- Text is inherently *unstructured* and *high-dimensional*.
- One of the major fields of application of machine learning methods.
- Fast-growing field. Many new techniques developed in industry.
- Increasingly used for applications in economics.

Working with text data

- Whole fields of research in different disciplines are devoted to this.
- Historically distinct fields like natural language processing, speech recognition, computational linguistics are merging in recent years.
- Extremely fast-paced development due to many practitioners.
- Research is commercially viable, well-performing implementations may not be public domain.
- Broad label: ‘Speech and language processing’ (Jurafsky & Martin 1999).

Text as data

- Text is inherently high-dimensional.
- Example: A sample of 30-word Twitter documents using only the 1,000 most common words in the english language ($w = 30$, $p = 1,000$).
- Unique representation p^w: About as many dimensions as atoms in the universe.

➥ Requires thinking about how text can be represented as data.

Text as data

- Order matters: $p^w = 1000^{30} = 1 \cdot 10^{90}$
- Order does not matter: $\left(\!\binom{p}{w}\!\right) = \binom{p+w-1}{w} = 5.79 \cdot 10^{57}$

Text as data

- Order matters: $p^w = 1000^{30} = 1 \cdot 10^{90}$
- Order does not matter: $\left(\!\binom{p}{w}\!\right) = \binom{p+w-1}{w} = 5.79 \cdot 10^{57}$
- Do these sentences convey the same meaning?
  - `He told only his mistress that he loved her.`
  - `He told his only mistress that he loved her.`
  - `He told his mistress only that he loved her.`
  - `He told his mistress that only he loved her.`
  - `He told his mistress that he only loved her.`
  - `He told his mistress that he loved only her.`
  - `He told his mistress that he loved her only.`
  - `He only told his mistress that he loved her.`
  - `Only he told his mistress that he loved her.`

Other challenges

- Human language is *highly ambiguous*.
  - `I ate pizza with friends.`
  - `I ate pizza with olives.`

Other challenges

- Human language is *highly ambiguous*.
  - `I ate pizza with friends.`
  - `I ate pizza with olives.`
- Human language is *highly variable*.
  - `I ate pizza with friends.`
  - `Friends and I shared some pizza.`

Other challenges

- Human language is *highly ambiguous*.
  - `I ate pizza with friends.`
  - `I ate pizza with olives.`

- Human language is *highly variable*.
  - `I ate pizza with friends.`
  - `Friends and I shared some pizza.`

➥ Requires thinking about how text can be represented as data.

➥ How do we encode meaning on the word- and document-level?

Terms and notation

- A *corpus* D is a set of *documents* $\{D_i : i = 1, \ldots, N\}$.
- We are seeking a matrix representation $\mathrm{C}_{N \times p}$ of D that can be used as *input* for statistical modeling.
- Rows of C correspond to documents, columns to input features (e.g. tokens, words), and $\mathrm{C} \in \mathcal{R}^{N \times p}$.
- We want to use C to model some target $\mathrm{V} = f(\mathrm{C})$.
- V may be observed or latent.

Language

- Different fields (Statistics, Econometrics, CS, ML, ...) employ different terms for the same objects.
- Dependent variable: Outcome, response, label, target, regressand, criterion, predicted variable, explained variable, ...
- Independent variable: Feature, predictor, input, covariate, control, regressand, explanatory variable, exposure variable, ...
- Tuning parameters are also commonly called hyperparameters outside statistics.
- Different focus: Predictive accuracy vs. consistency and correct inference.

Basic analysis steps

1. Represent raw text D as a numerical array C.
2. Map C to predicted values of (potentially unknown) target V.
3. (Use $\hat{V}$ in subsequent descriptive or causal analysis, $Y = f(\hat{V})$.)

Analysis summary

1. Represent raw text D as a numerical array C.

- Pre-processing: Researcher must impose some preliminary restrictions to reduce the dimensionality of the data.
- No current technique can deal with 1000^{30}-dimensional Twitter data.
- In most cases, elements of C are counts of *tokens* (e.g. words, phrases, pre-defined features).
- Other approaches leverage prior information about the structure of language before any analysis to reduce dimensionality.
- Dimension likely to be large, possibly $p \gg n$.

Analysis summary

2. Map C to predicted values of (potentially unknown) target V.

- Dimension smaller, $k \ll n$, often just a vector.
- Involves application of (high-dimensional) statistical methods.
- V may be *observable* or *latent*, application of statistical models involves either *supervised* or *unsupervised* learning.
- Classical example: data is the text of emails, V is an indicator for whether the email is spam, prediction $\hat{V}$ determines whether to flag email as spam.

Analysis summary

2. Map C to predicted values of (potentially unknown) target V.

- Typically V is either
  - a *direct* outcome/predictor (sensible on its own, e.g. spam)
  - an *intermediate* low-dimensional representation of the text data.
- Examples: Sentiment prediction, predicting flu outbreaks from google searches, grouping texts by topic, stock prices, political slant.
- ...or just numbers in a denser vector space of smaller dimension than C.
- Distinction between *representing* text and *analyzing* text is blurry.

Analysis summary

3. Use $\hat{V}$ in subsequent descriptive or causal analysis.

- Goal is often prediction rather than causal inference; the interpretation of the mapping from V to $\hat{V}$ is not usually of interest.
- Aim in social science is often to use $\hat{V}$ to infer causal relationships or structural parameters.
- Examples:
  - Interested in the effect of a medical intervention. Based on a vector space representation of the medical dossiers, you identify a set of patients with similar conditions to the intervention group to be used as a control group.
  - Identify topics discussed in parliament. Find out how discussions shift in times of crisis.
- Optional and application-specific. Not covered in detail here.

Representing text as data

- Humans read text in context, not vectors of binary variables or sequences of unrelated tokens.
- We interpret words in light of other words, extracting meaning from a text as a whole.
- The relevant information is mixed with a lot of irrelevant information.

➥ Text analysis ignores much of this complexity.

➥ Extracting the relevant information for a task is key.

➥ Recent advances in context-sensitive NLP methods (e.g. translations) have largely come about because of better *representations*, not models.

Representing text as data

- Typical simplifications when constructing C:
  (a) divide text into individual documents,
  (b) reduce the number of language elements considered,
  (c) limit the extent to which we encode dependence among elements within documents.
- Map raw text D to a numerical array C.
- Typically, a row c_i of C is a vector with each element indicating the presence or count of a particular language token j in document i.

Defining a document

- Divide raw text D into individual documents $\{D_i\}$.
- Level often determined by the level at which V is defined.
- Choice not always clear. Finer partitions ease computation at the cost of limiting the dependence that can be captured.
- No theoretical guidance.
- Most methods treat documents as independent.
- For social science applications, documents typically matter.
- For many language modeling applications, individual documents matter less.

Representing text as data

Pre-processing and feature selection

Pre-processing and feature selection

- Pre-processing common to restrict the set of language elements considered.
- Not all methods require or benefit from this.
- For now, we focus on *token-based* representations: Elements of C are counts c_{ij} of some language tokens.
- Feature selection and pre-processing ease computational burden and remove noise.

Common pre-processing

- Cleaning: Remove elements that do not belong (e.g. HTML tags)
- Capitalization: Transform all text to lower case.
- Strip elements from the raw text that are not words, removing punctuation, proper names, numbers, etc.
- Use lemmatization/stemming to homogenize the raw text.
- Remove subsets of words that are very common or very rare.

Lemmatization

- Lemmatization is the task of determining that two words have the same root.
- Replace words with their root: '*economic*', '*economics*', '*economically*' are replaced by '*economic*'.
- Difficult, requires complicated morphological parsing algorithms.
- Morphological parser: '*cats*' parsed into two morphemes '*cat*' and '*s*'.
- Non-trivial: A Spanish word like '*amaren*' ('if in the future they would love') would need to be parsed into the morphemes *amar* ('to love'), *3PL*, and *future subjunctive*.

Stemming

- Stemming is a naive version of morphological analysis.
- Cut off word-final affixes based on a series of rewrite rules.

`ATIONAL`	$\longrightarrow$ `ATE`	(e.g., relational $\longrightarrow$ relate)
`ING`	$\longrightarrow \epsilon$ *if stem contains vowel*	(e.g., motoring $\longrightarrow$ motor)
`SSES`	$\longrightarrow$ `SS`	(e.g., grasses $\longrightarrow$ grass)
...	$\longrightarrow$ ...	

Stemming

- Different stemming tools available, Porter (1980) has become standard for English and performs well in practice.
- Algorithm based on sequential rules. Errors of both over- and under-generalizing occur.
- Example input/output:

```
This was not the map we found in Billy Bones's chest, but
an accurate copy, complete in all things-names and heights
and soundings-with the single exception of the red crosses
and the written notes.

Thi wa not the map we found in Billi Bone s chest but an
accur copi complet in all thing name and height and sound
with the singl except of the red cross and the written note
```

Direct feature selection: Stop words

- Remove subsets of words that are very common or very rare.
- ‘Stop words’ are very common words, e.g. articles (‘the’, ‘a’), conjunctions (‘and’, ‘or’), forms of the verb ‘to be’, and more.
- Important for the grammatical structure but conveying little meaning on their own.
- Removal based on pre-defined lists is common practice.

Direct feature selection: Frequency filtering

- Very common words are often removed because they carry very little meaning on their own.
- Very rare words do convey meaning, but the associated computational cost often exceeds their diagnostic value.
- To reduce sparsity, terms are frequently filtered as well.
- Common practice is to exclude all terms that occur fewer than k times or in fewer than k% of documents for some arbitrary small k.

Filtering by tf-idf

- Term-frequency-inverse-document-frequency (*tf-idf*) is a combined approach to filtering.
- *Term frequency*: The (normalized) count of occurences of a word in a document.
- *Inverse document frequency*: The log of one over the share of documents containing the word.

$$tf_{ij} \times idf_j = \frac{c_{ij}}{\sum_j c_{ij}} \times \log\left(\frac{N}{\sum_i \mathbb{1}\{c_{ij} > 0\}}\right)$$

Filtering by tf-idf

- Filter words with low *tf-idf* scores below some cutoff.
- Rare words have low scores because of low *tf*.
- Very common words that appear in most documents have low scores due to low *idf*.
- Higher scores for words that are occur frequently in some documents but not in others.
- These words are likely to be distinctive and informative about the content of documents.
- tf-idf scores can also be used as weights.

Issues

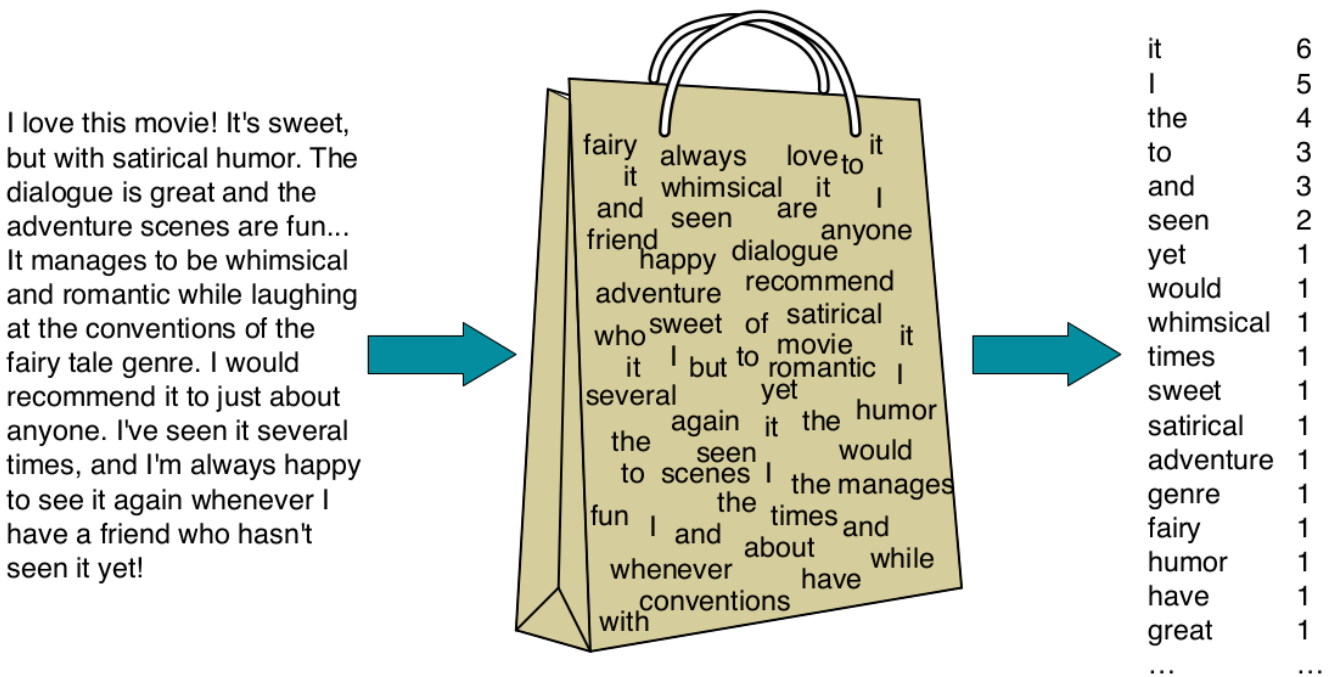
- Cleaning helps reducing the number of unique language elements and the dimensionality of the data.
- Provides computational benefits and is often key to getting interpretable model fits (e.g. in topic modeling).
- But requires careful decisions about the elements likely to carry meaning in the particular application.
- 'One researcher's stop words are another's subject of interest.'
- Context matters. Filtering may imply loss of information.
- Drop numerals from 'the first 100 days' or 'September 11'? How about online communication, e.g. :-) ?

Representing text as data

N-grams

N-grams

- Limit dependence among language elements to get a tractable representation.
- Simplest approach: Bag-of-words/1-grams. Ignore word order.
- c_i is a vector of the length of the vocabulary contained in D and the elements c_{ij} are counts of occurence.



N-grams

Good night, good night! Parting is such sweet sorrow.

`good night good night part sweet sorrow`

1-grams
$c_{ij} = 2$ for $j \in \{good, night\}$
$c_{ij} = 1$ for $j \in \{part, sweet, sorrow\}$
$c_{ij} = 0$ for all other words in the vocabulary

2-grams
$c_{ij} = 2$ for $j \in \{good.night\}$
$c_{ij} = 1$ for $j \in \{night.good, night.part, part.sweet, sweet.sorrow\}$
$c_{ij} = 0$ for all other possible 2-grams

Representation concepts: Term-document-matrix

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

Figure 15.1 The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

Figure 15.2 The term-document matrix for four words in four Shakespeare plays. The red boxes show that each document is represented as a column vector of length four.

Vector representation: Term-document-matrix

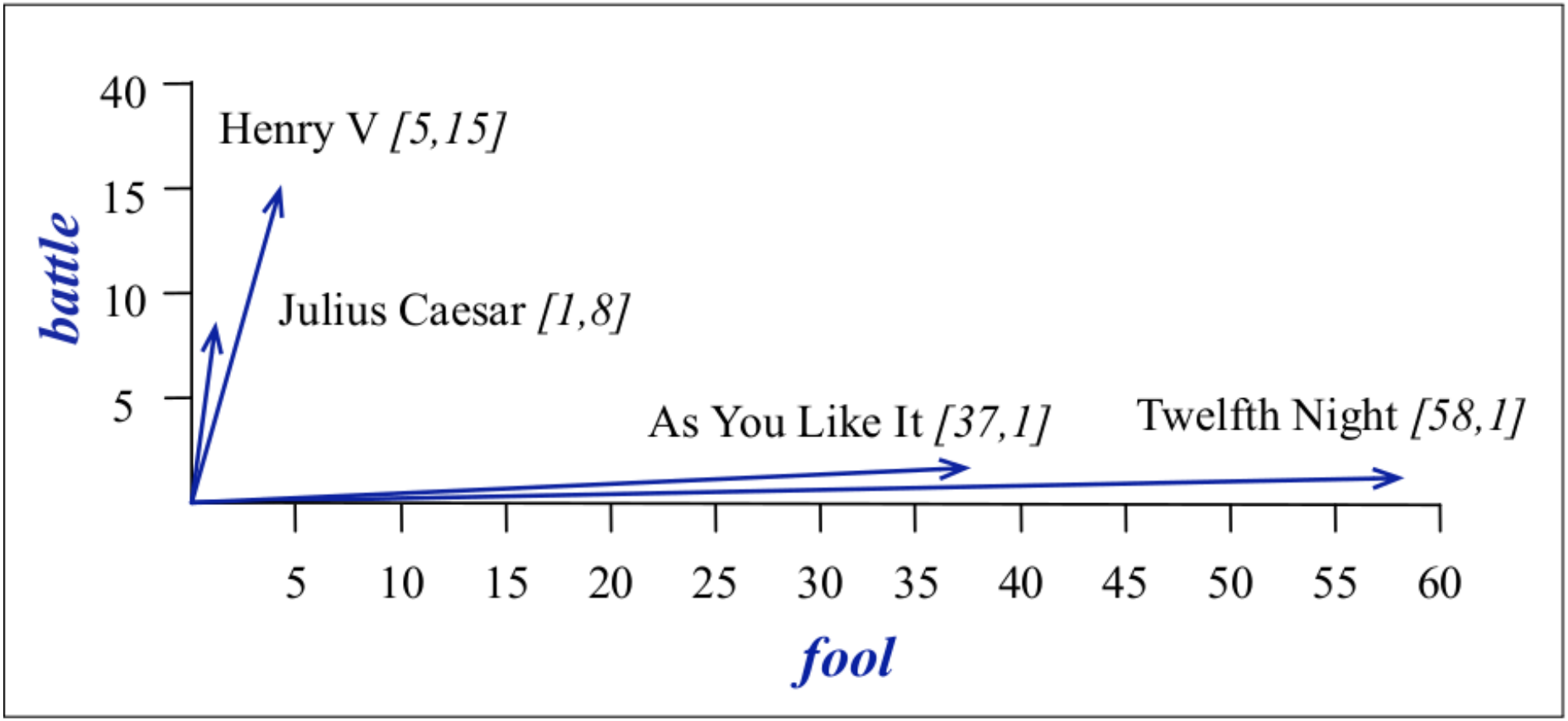


Figure 15.3 A spatial visualization of the document vectors for the four Shakespeare play documents, showing just two of the dimensions, corresponding to the words *battle* and *fool*. The comedies have high values for the *fool* dimension and low values for the *battle* dimension.

N-grams

- n-grams of order $n > 1$ yields data that captures a limited amount of dependence between words.
- n-gram counts sufficient for estimation of an n-order homogenous Markov model across words.
- Model assumes word choice only depends upon the previous n words.
- Markov model applications: Speech recognition, spelling corrections, automatic sentence completion ...

Markov chains

- Consider a language model seeking to predict the probability of observing the word $t+1$ given the previous t words.

$$p(w_{t+1}|w_1, w_2, \ldots, w_t)$$

- Using the chain rule, we can compute the probability of observing an entire sentence.

$$p(w_1, w_2, \ldots, w_t) = p(w_1)p(w_2|w_1)\cdots p(w_t|w_{t-1}, \ldots, w_1)$$

- Can be computed using maximum likelihood estimate.

$$p(x_{t+1}|x_1, x_2, \ldots, x_t) = \frac{count(x_1, x_2, \ldots, x_t, x_{t+1})}{count(x_1, x_2, \ldots, x_t)}$$

➥ Computing this statistic is unrealistic. We are unlikely to observe enough data to obtain realistic counts for a sequence of t words for any nontrivial value of t.

Markov chains

- Invoke the Markov assumption: Probability of observing a word at a given time is only dependent on the $n-1$ previous words, e.g.

$$p(x_{t+1}|x_1, x_2, \ldots, x_t) = p(x_{t+1}) \quad \text{(unigram)}$$
$$p(x_{t+1}|x_1, x_2, \ldots, x_t) = p(x_{t+1}|x_t) \quad \text{(bigram)}$$
$$p(x_{t+1}|x_1, x_2, \ldots, x_t) = p(x_{t+1}|x_t, x_{t-1}) \quad \text{(trigram)}$$

- The probability of a sentence is then

$$p(w_1, w_2, \ldots, w_t) = \prod_{i=1}^{t} p(w_i) \quad \text{(unigram)}$$
$$p(w_1, w_2, \ldots, w_t) = p(w_1) \prod_{i=2}^{t} p(w_i|w_{i-1}) \quad \text{(bigram)}$$
$$p(w_1, w_2, \ldots, w_t) = p(w_1) \prod_{i=2}^{t} p(w_i|w_{i-1}, w_{i-2}) \quad \text{(trigram)}$$

N-grams

- N-gram models for $n > 1$ allow richer structure where simple words may be insufficient to capture patterns of interest, e.g. partisan overtones in 'death tax' or 'tax break'.
- But: dimension of c_i increases exponentially with n. More than 3-grams is rarely used. Return small relative to cost.
- Begin with bag-of-words, evaluate whether moving to bi-gram/tri-gram is worthwile.

Practical considerations

- Subsets/transformations of the term-document matrix often used as C.
- N-gram representation increases the dimension of features substantially.
  - Typical vocabulary: $\approx$ 35,000 (average person).
  - All possible 2-Grams in the English dictionary: 171,476 *choose* 2.
- Reduce sparsity by filtering low frequency n-grams (or low tf-idf, PPMI, ...).
- Other dimensionality reduction techniques can also be applied (PCA/SVD, factor analysis, penalized regression, ...).
- Initial feature selection necessarily aggressive—may not be inoccous.

Representation and model choice

- The choice of representation and statistical model are often linked. Some models are specific to the structure of text, others are not.

➥ Focus first on token-count/n-gram input and suitable statistical models.

- Other representation approaches lead to less sparse inputs which can be used with a wider class of models.

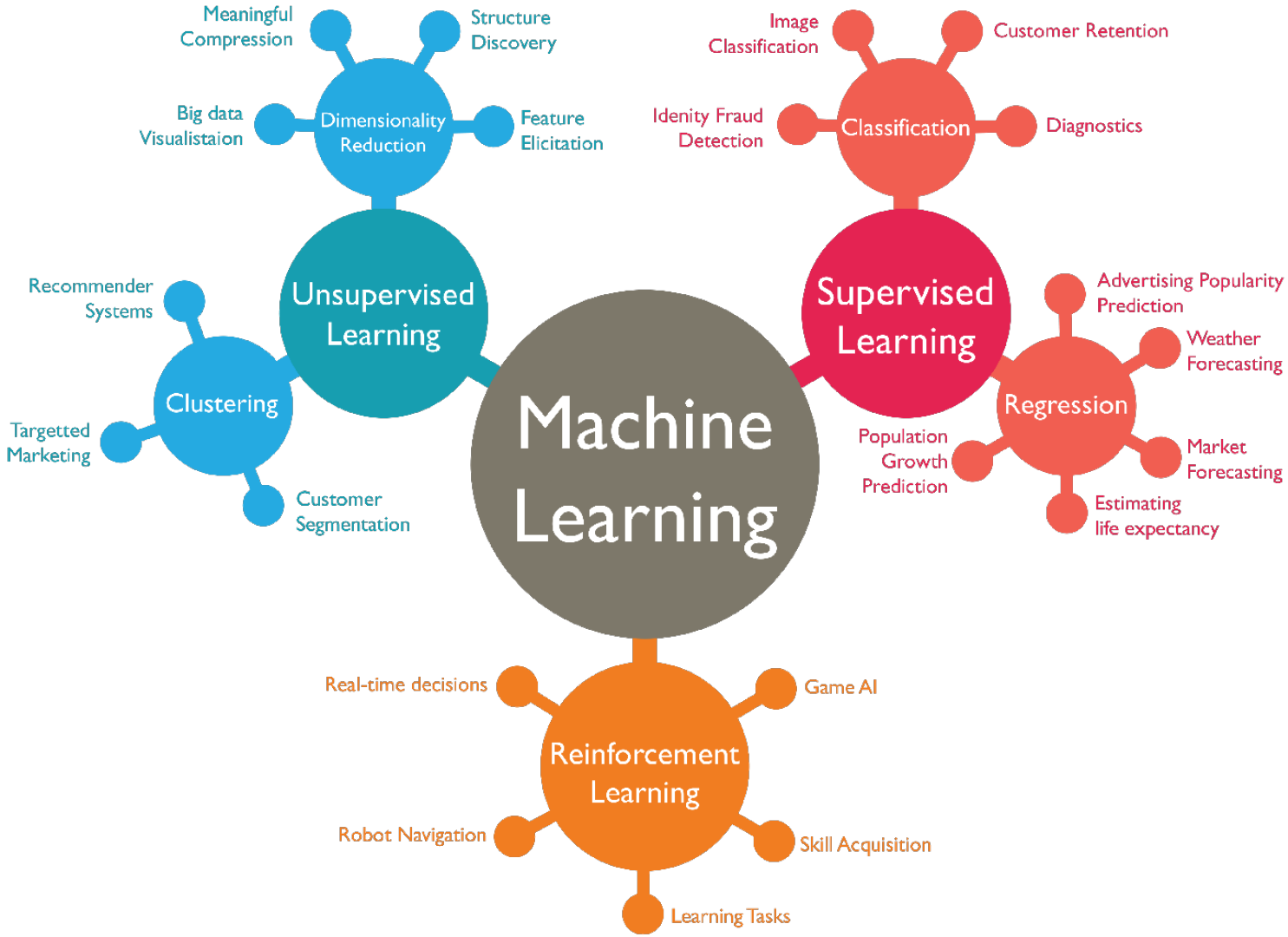
➥ Focus on these in the second part of the lecture.

Statistical methods

Statistical methods

Introduction

Elements of ML: Supervised and unsupervised learning



Supervised vs. unsupervised learning

Supervised learning

- Outcome/dependent variable is *observed* (data is 'labeled').
- Data for Y and X is available and we want to learn $\hat{Y} = \hat{f}(X)$.
- *Classification* when Y is discrete.
- *Regression* when Y is continuous.

Unsupervised learning

- Outcome/dependent variable is *latent*.
- Only data for X is available and we want to learn about its structure.
- *Clustering:* Partitioning data into (homogenous) groups based on X.
- *Dimensionality reduction:* Represent data in smaller dimension, input transformation (e.g. PCA, word2vec, ...).

Supervised learning

- In supervised ML the goal is prediction: Learn $\hat{Y} = \hat{f}(X)$ to predict Y.
- We care about accurate predictions $\hat{Y}$ for Y on *new* data.
- In contrast to causal inference, we do not interpret parameters of $\hat{f}(X)$.
- Consistent estimation of parameters in $\hat{f}(X)$ does not matter.
- Fit $\hat{f}(X)$ to maximize *out-of-sample* predictive performance.
- Out-of-sample performance is assessed using model fit criteria.
  - Regression: Mean-square-error
  - Classification: Accuracy, AUC, sensitivity, specificity, ...
  - AIC, BIC, ...

Supervised learning in one slide

Objective function

$$\min \sum \underbrace{L(Y, f(X))}_{\text{loss function}} \text{ over } \underbrace{f \in F}_{\text{function class}} \quad \text{s.t.} \quad \underbrace{R(f) \leq c}_{\text{complexity restriction}}$$

- Loss function $L(\cdot)$: quadratic error loss, absolute, ...
- Function class F: linear (e.g. penalized linear regression), non-linear (e.g. penalized logistic regression), non-parametric (e.g. trees, neural nets).
- Complexity restriction $R(f)$: size of penalty, depth of tree, number of hidden layers, ...
- Complexity level c: tuning parameter for max. complexity.
- Set tuning parameters s.t. out-of-sample predictive performance is maximized.

Assessing predictive performance

Regression

- Mean-squared-error (MSE):

$$MSE_{\hat{Y}} = \mathbb{E}[(\hat{Y} - Y)^2] = \underbrace{\mathbb{E}[(\hat{Y} - \mathbb{E}[\hat{Y}])^2]}_{\text{Variance}} + \underbrace{(\mathbb{E}[\hat{Y} - Y])^2}_{\text{Bias}^2}$$

Classification

- Misclassification rate:

$$\frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(Y \neq \hat{Y})$$

Classification vs. regression

- In practice, classification applications much more common than regression.
- Some models do not deliver discrete probabilities (e.g. trees vs. logistic regression). Obtain predicted class from probability.
- In many applications, false positive and false negative prediction errors are not weighed equally.
- Check contingency table of predicted vs. actual values ('confusion matrix'). Various model performance statistics can be computed from it.
- Target other measures (sensitivity, specificity) to limit type-I or type-II errors.

Confusion matrix

Predicted condition \ True condition	Positive	Negative		
Positive	True positive	False positive (Type-I error)	Positive pred. value $\frac{TP}{TP+FP}$	False discovery rate $\frac{FP}{TP+FP}$
Negative	False negative (Type-II error)	True negative	False omission rate $\frac{FN}{FN+TN}$	Negative pred. value $\frac{TN}{FN+TN}$
	True positive rate $\frac{TP}{TP+FN}$	False positive rate $\frac{FP}{FP+TN}$		
	False negative rate $\frac{FN}{TP+FN}$	True negative rate $\frac{TN}{FP+TN}$		

Prevalence

Condition positive/total population, $\frac{TP+FN}{TP+FP+TN+FN}$

Accuracy

(True positive + true negative)/total population, $\frac{TP+TN}{TP+FP+TN+FN}$, 1 - misclassification rate

Confusion matrix

Predicted condition \ True condition	Positive	Negative		
Positive	True positive	False positive (Type-I error)	Positive pred. value *(precision)*	False discovery rate
Negative	False negative (Type-II error)	True negative	False omission rate	Negative pred. value
	True positive rate *(sensitivity, recall, power)*	False positive rate *(fall out)*		
	False negative rate *(miss rate)*	True negative rate *(specificity, selectivity)*		

Prevalence

Condition positive/total population, $\frac{TP+FN}{TP+FP+TN+FN}$

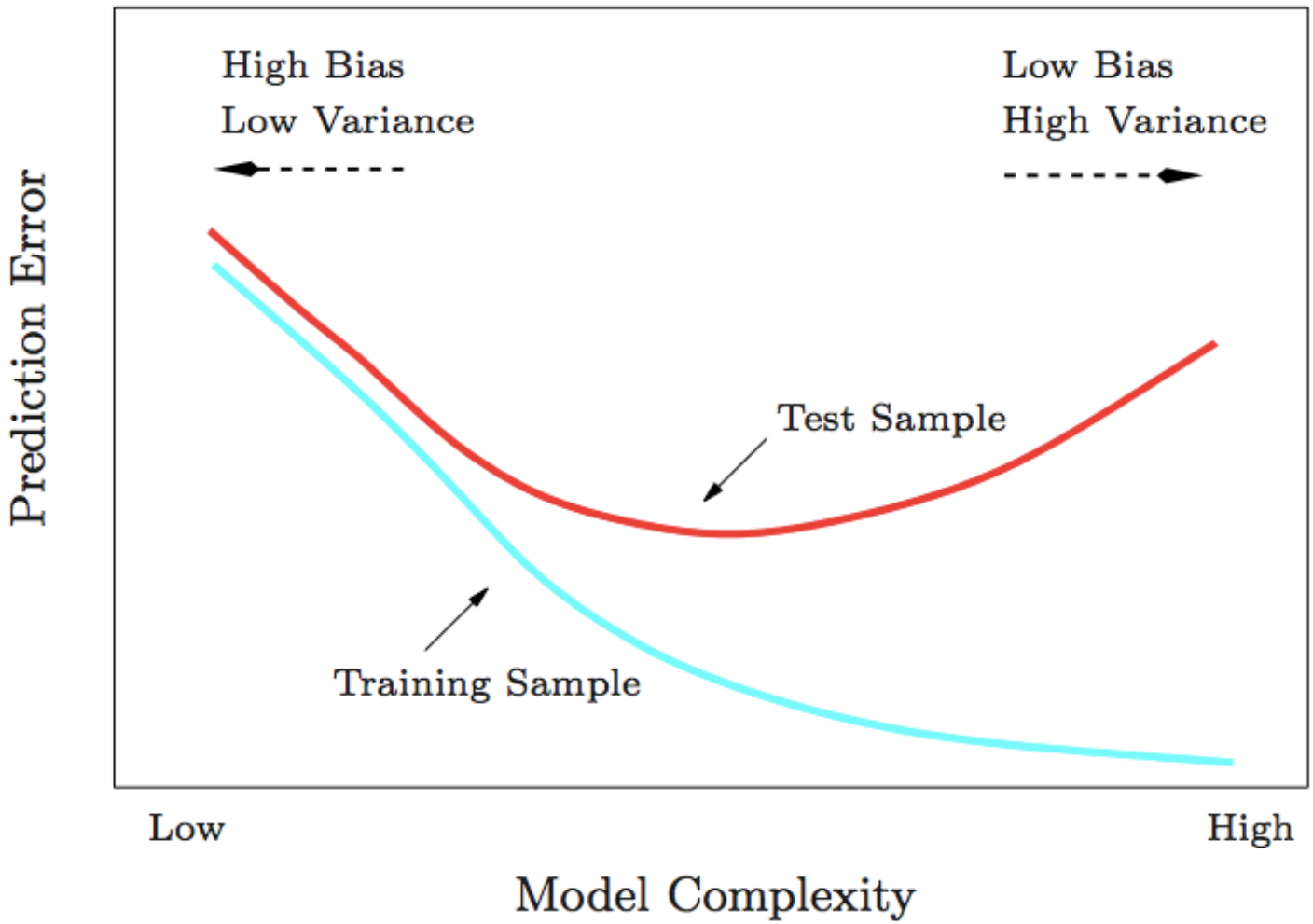
Accuracy

(True positive + true negative)/total population, $\frac{TP+TN}{TP+FP+TN+FN}$, 1 - misclassification rate

Model training and testing

- Out-of-sample assessment is important due to overfitting.
- Data is split into training sample for estimation, test/hold-out sample for assessment.
- Typically three splits: training, validation, test/hold-out.
- Cross-validation (or other resampling methods) can be used to tune model parameters.
- Resampling is used to pick the optimal tuning parameters (the best performing model within a class of models).
- True test sample always necessary for comparative model assessment.

Overfitting



K-fold cross-validation

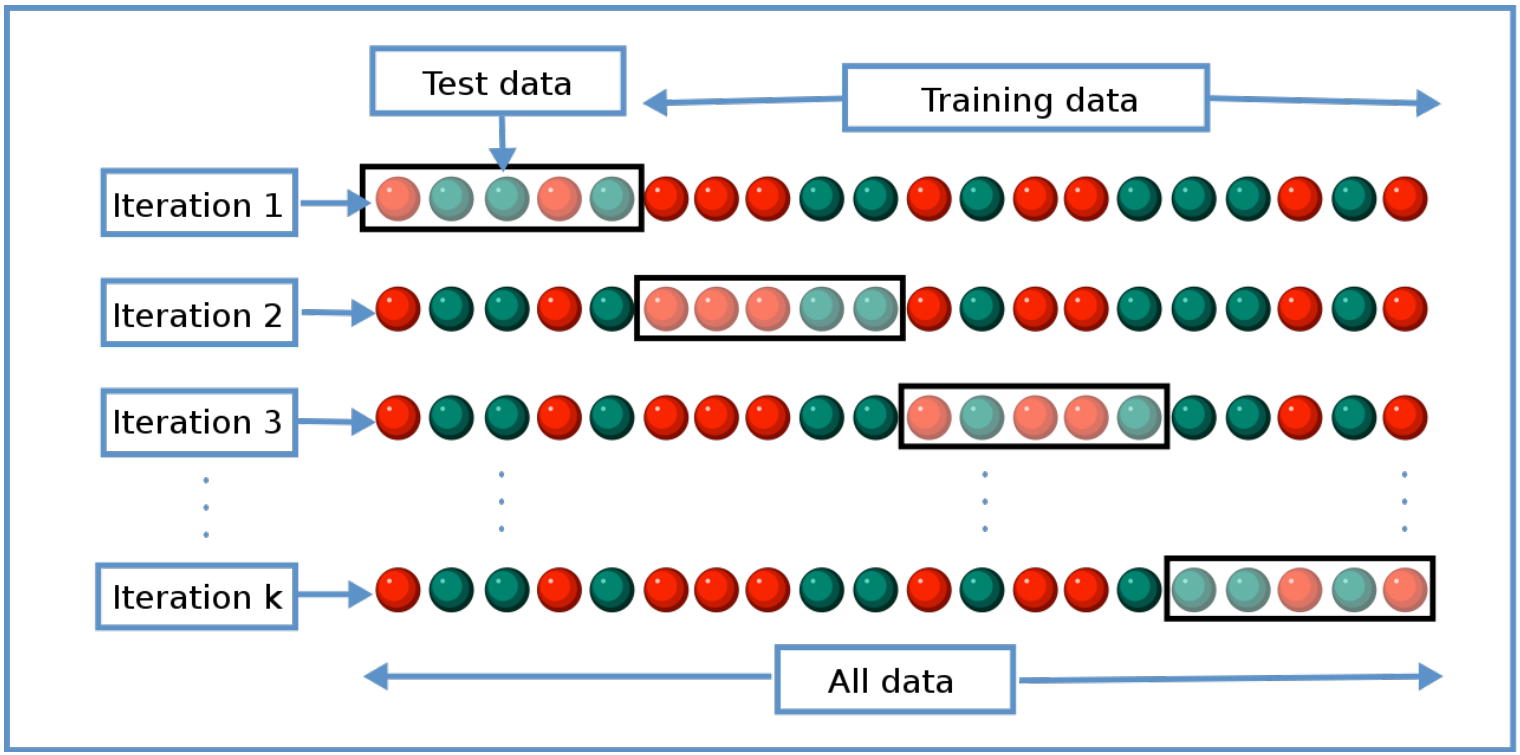
k-fold CV

1. Randomly divide observations into k groups (folds) of approx. equal size.
2. The first fold is treated as a validation fold, and the model is fit on the remaining $k - 1$ folds.
3. Compute the fit statistic (MSE) on the observations in the held-out fold.
4. Repeat 2. and 3. k-times for each fold.
5. Compute the k-fold CV estimate of the MSE by averaging the values

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} \text{MSE}_i$$

[6.] [Repeat 2.–5. for different tuning parameter sets (e.g. grid search). Pick the model with the smallest estimate of $CV_{(k)}$.]

K-fold cross-validation



K-fold cross-validation

- Number of folds typically 5–20, $k = 10$ commonly used.
- Special case: Leave-one-out-cross-validation. More computationally expensive and does not perform better.
- Van der Vaart et al. (2006) show that cross-validation tuning approximates the optimal complexity.
- Sometimes ad-hoc extensions (e.g. one-sd rule for the Lasso).
- Alternatives to CV:
  - Information criteria
  - In rare cases, there is theoretical guidance on the optimal choice of tuning parameters and CV is not needed (e.g. Belloni et al. 2012).

Evaluating model performance

- Why not use CV exclusively?
- With a lot of tuning parameters, you may end up overfitting even with CV.
- Always use a *true* hold-out sample to evaluate model performance.
- You want to assess your final model on data that has never been used for training.
- Typically 5-20% of the data, depending on size.
- This is also best for assessing comparative model performance.

Statistical models for text data

- Statistical methods used to analyze text are closely related to methods used to analyze high-dimensional data in other domains.
- Lasso/penalized regression applied more or less as in other settings.
- Other methods (e.g. topic models) have been adapted to the specific structure of text data.
- Unlike in classical econometrics, *unsupervised* and *generative* methods are important with text data.

Notation

- Methods for mapping the document-token matrix C to predictions $\hat{\mathrm{V}}$ of an attribute V.
- Data may be partitioned such that C^{train} (dim. $n^{train} \times p$) collects rows for which V^{train} (dim. $n^{train} \times k$) of V is observed.
- p is the number of input features, k the number of attributes to predict.
- Matrix X may denote an input matrix which is a transformation of C, either of the same dimension or with a reduced number of features/columns ($n \times p_r$, with $p_r \leq p$).
- Attributes in v can be observable quantities (flu cases, movie review, unemployment rate) or latent (topics being discussed in politics or news).

Statistical methods

Four broad classes of models connecting counts c_i to attributes v_i:

1. Dictionary-based methods (no statistical model).
2. Discriminative models (modelling $p(v_i|c_i)$ directly).
3. Generative models (modeling $p(c_i, v_i) = p(c_i|v_i)p(v_i)$)
4. Distributed language models leveraging richer text representation than token counts (dimension reduction).

Discriminative vs. generative models

- *Discriminative* models learn the posterior $p(v_i|c_i)$ directly.
- *Generative* models learn a model of the joint probability $p(c_i, v_i)$ and use Bayes' rule to calculate the posterior.
- Discriminative models map $c \mapsto v$, generative models map $v \mapsto c$.
- Discriminative model are mostly non-parametric (models with a non-fixed amount of parameters are considered non-parametric).
- Generative models are mostly parametric since they attempt to model the underlying joint distribution.
- Both can be *supervised* or *unsupervised*.

Discriminative vs. generative models

- Generative models not common in traditional econometric applications.
- Usually advisable to solve the classification/regression problem directly (learning a direct mapping from $x \mapsto y$),
- ... rather than solving a more general problem as an intermediate step.
- Generative models are more common with text data and have some justification due to the structure of language.
- Typically rely on a strong conditional independence assumption of features. Even though this assumption is often implausible, they perform well in a variety of tasks.
- Partially because flexible discriminative methods do not always perform well with very sparse data.

Statistical methods

Dictionary methods

Dictionary methods

- No statistical inference at all.
- Specify $\hat{v}_i = f(c_i)$ for some known function $f(\cdot)$.
- The most common method in the social science literature so far.
- Popular for sentiment analysis/detection.
- $f(\cdot)$ defined based on pre-defined or ad-hoc categorization, e.g. using a dictionary of word lists associated with sentiment categories.

Dictionary methods

- Pre-compiled dictionaries for many different contexts are now available.
  - Positive/neutral/negative sentiment, argumentativeness, positive/negative effect events, ...
- Some of these are simple regex match lists, others also parse sentence structure (e.g. subject/verb/object recognition).
- Many 'standard' dictionaries are accessible in R.

Developing your own dictionary

- In many settings pre-compiled dictionaries are not helpful (e.g. due to specificity or language).
- Easy to implement your own dictionary mapping using search strings or regular expressions.
- Be methodical about implementation. Document your thought process.
- Consider type-I and type-II errors when designing your own scheme. Always check some examples. Accounting for 'exceptions to the rule' can get tedious and complicated quickly.
- Validate your dictionary. Have others conduct the same task in parallel.

Statistical methods

Text regression

Text regression

- Regression is a familiar modeling approach.
- Start with a model for $p(\mathsf{v}_i|\mathsf{c}_i)$. Training data is available.
- Predicting v_i from c_i is a standard regression problem.
- But: OLS is infeasible if $p \geq n^{train}$ (infinitely many solutions).
- High dimensionality of c_i requires appropriate techniques.
- Best subset selection is too costly given the dimension of c.

➥ Penalized linear (index) regression popular for text analysis.

Penalized regression and shrinkage

- Penalized linear (index) model:

$$E(\mathsf{v}_i|\mathsf{x}_i) = f(\boldsymbol{\eta}_i)$$

  where $\boldsymbol{\eta}_i = \alpha + \mathsf{x}_i'\boldsymbol{\beta}$ is a linear index, x_i a known transformation of the text token counts c_i and $f(\cdot)$ some link function.

- Link function $f(\cdot)$: Typically identity (linear regression) or logit (logistic regression) for binary outcomes.

➥ Add additional penalty term to the objective function to shrink the size of coefficients (and reduce the number of tokens included in the model).

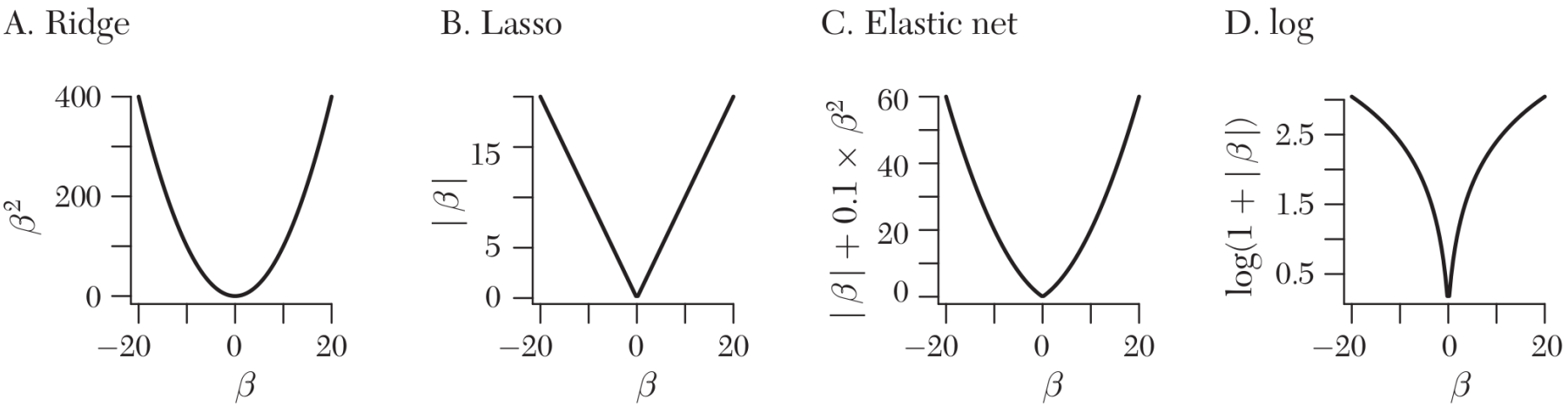
Penalized regression

- Penalized linear regression solves

$$\arg\min_{\alpha,\boldsymbol{\beta}} \left\{ \sum_i (v_i - \alpha_i - \mathrm{x}_i'\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} \kappa_j(\beta_j) \right\}$$

- λ controls the size of the penalty for coefficient size.
- κ is a penalty function. Common penalties are $\sum_1^p \beta_j^2$ (L_2-penalty, ridge regression) and $\sum_1^p |\beta_j|$ (L_1-penalty, Lasso).

L_1 penalized regression



- Unlike for ridge regression, the nature of the L_1 penalty causes some coefficients to shrink to zero exactly.
- L_1 penalized linear or logistic regression ('Lasso', Tibshirani 1996) is the best choice for text regression.
- For simple text regression tasks seldom possible to do much better.

L_1 penalized text regression

- Lasso deviance objective function:

$$\min \left\{ \ell(\alpha, \boldsymbol{\beta}) + n\lambda \sum_{j=1}^{p} \omega_j |\beta_j| \right\}$$

- $\ell(\alpha, \boldsymbol{\beta})$ is an unregularized objective proportional to the negative log-likelihood $\log p(\mathsf{v}_i|\mathsf{c}_i)$.
- Linear (gaussian) regression: $\ell(\alpha, \boldsymbol{\beta}) = \sum_i (\mathsf{v}_i - \boldsymbol{\eta}_i)^2$
- Logistic (binomial) regression: $\ell(\alpha, \boldsymbol{\beta}) = -\sum_i (\boldsymbol{\eta}_i \mathsf{v}_i - \log(1 + e^{\boldsymbol{\eta}_i}))^2$ if $\mathsf{v}_i \in \{0, 1\}$.

L_1 penalized text regression

- Lasso deviance objective function:

$$\min \left\{ \ell(\alpha, \boldsymbol{\beta}) + n\lambda \sum_{j=1}^{p} \omega_j |\beta_j| \right\}$$

- Common strategy to set ω_j such that the penalty cost for each coefficient is scaled by the sample standard deviation of that covariate.
- 'Rare feature upweighting' is good practice: Each covariate corresponds to a specific text token, and rare words are often most useful in differentiating between documents.
- Model selection via cross-validation, AIC, BIC.

Token count transformation

- Penalized linear (index) model:

$$E(\mathsf{v}_i|\mathsf{x}_i) = f(\alpha + \mathsf{x}_i'\boldsymbol{\beta})$$

- x_i is a transformation of the token counts c_i.
- Common transformations for c_i are the identity $\mathsf{x}_i = \mathsf{c}_i$, normalization by document length $\mathsf{x}_i = \mathsf{c}_i/m_i$ with $m_i = \sum_j c_{ij}$ or just an occurence indicator $c_{ij} = \mathbb{1}(c_{ij} > 0)$.
- Choice depends on application and interpretation.
- Identity or binary indicator (especially after filtering) are reasonable defaults.

Other text regression models

- Penalized regression most widely applied tool.
- Previously, people often just applied PCA to the term-document matrix and used some of the principal components for regression ('Latent Semantic Analysis').
- Bayesian regression methods available.
- Tree-based methods can be used, but are less useful with high-dimensional text-based inputs.
- Benefits of trees (nonlinearity, higher-order interaction detection) cannot be reaped with sparse binary inputs. Resampling is costly.
- May be useful for a final prediction step after a dimension reduction from a generative model or a distributed language model.

Statistical methods

K-means clustering

K-means clustering

- Simple approach for partitioning a data set into K distinct, non-overlapping clusters (Steinhaus 1956, MacQueen 1967, Forgy 1965).
- Specify desired number of clusters K ex ante, then assign each observation to exacly one cluster.
- Example: Categorize documents into (latent) groups.

K-means clustering

- Simple approach for partitioning a data set into K distinct, non-overlapping clusters (Steinhaus 1956, MacQueen 1967, Forgy 1965).
- Specify desired number of clusters K ex ante, then assign each observation to exacly one cluster.
- Example: Categorize documents into (latent) groups.
- Let $I_1, \ldots, I_K$ denote sets containing the indices of the observations in each cluster. Sets satisfy
  1. $I_1 \cup I_2 \cup \cdots \cup I_K = \{1, \ldots, n\}$
     Each observation belongs to at least one cluster.
  2. $I_k \cap I_{k'} = \varnothing$ for all $k \neq k'$
     No observation belongs to more than one cluster.

K-means clustering

- Objective: Minimize total within-cluster variation $W(I_k)$.

$$\min_{I_1,\ldots,I_K} \left\{ \sum_{k=1}^{K} W(I_k) \right\}$$

- Variation metric: Squared Euclidean distance.

$$\min_{I_1,\ldots,I_K} \left\{ \sum_{k=1}^{K} \frac{1}{|I_k|} \sum_{i,i' \in I_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

- Difficult to solve exactly. K^n ways to partition n observations into K clusters. Complexity $O(n^{pk+1})$.
- Simple algorithm delivers *local* optimum.

K-means clustering: Algorithm

K-means clustering

1. Randomly assign each observation to a cluster, a number from 1 to K.
2. Iterate until assignments stop changing:
   (a) For each of the K clusters, compute the cluster centroid. The k-th cluster centroid is the vector of the p feature means for the observations in the k-th cluster.
   (b) Assign each observation to the cluster whose centroid is closest (defined using the L^2-norm).

K-means clustering: Algorithm

- Works because $(1/|I_k|)\sum_{i,i' \in I_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i\in I_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2$.
- Optimization problem can be reduced to

$$\min_{I_1,\ldots,I_K}\left\{\sum_{k=1}^{K}\sum_{i\in I_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{jk})^2\right\} = \min_{I_1,\ldots,I_K}\left\{\sum_{k=1}^{K}\|\mathbf{x}-\bar{\mathbf{x}}_k\|^2\right\}$$

- Reallocating observations can only improve objective until a local optimum is reached.
- Time complexity is $O(nkpi)$, where i is the number of iterations needed until convergence.
- Iterate between *assignment* and *update* step (simplified EM algorithm).

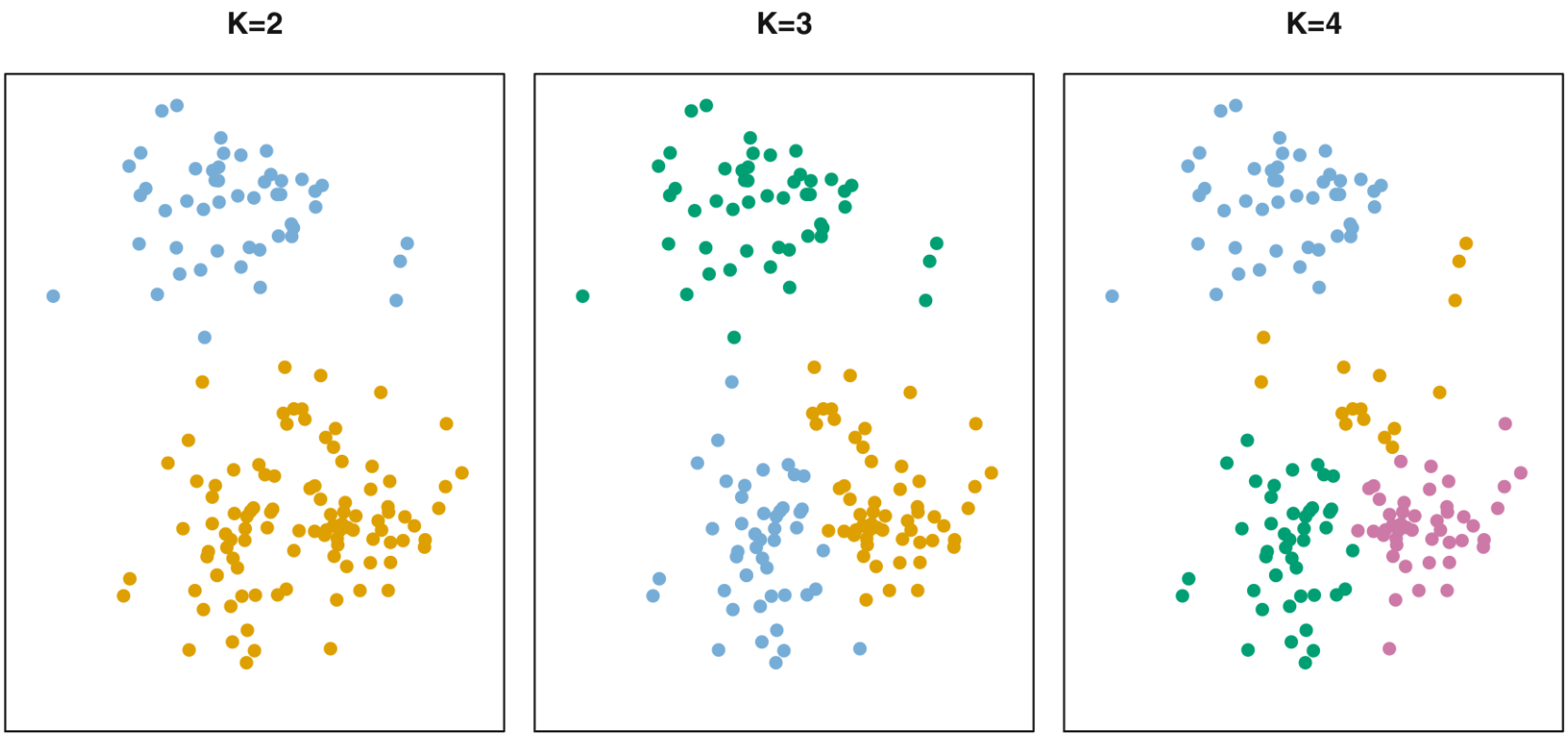
K-means clustering: Algorithm

K-means clustering

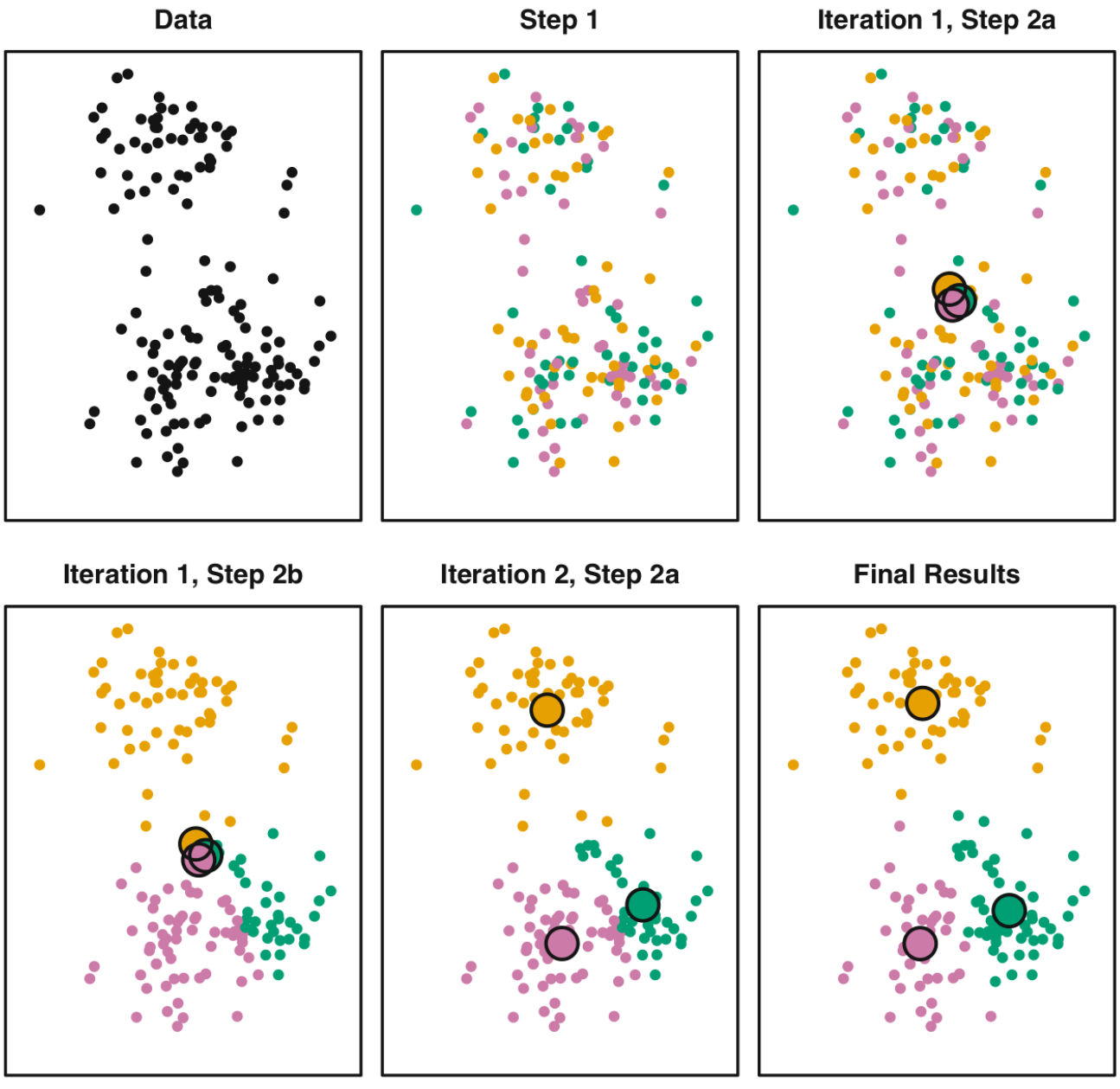
1. Randomly initiate cluster centroids $\bar{x}_{kj}$ for clusters $k = 1, \ldots, K$ and features $j = 1, \ldots, p$.
2. Iterate until assignments stop changing:
   (a) Assign each observation to the cluster whose centroid is closest (defined using the L^2-norm).
   (b) For each of the K clusters, compute the cluster centroid. The k-th cluster centroid is the vector of the p feature means for the observations in the k-th cluster.

➥ Iterate between *assignment* and *update* step (simplified EM algorithm).

K-means clustering: Example

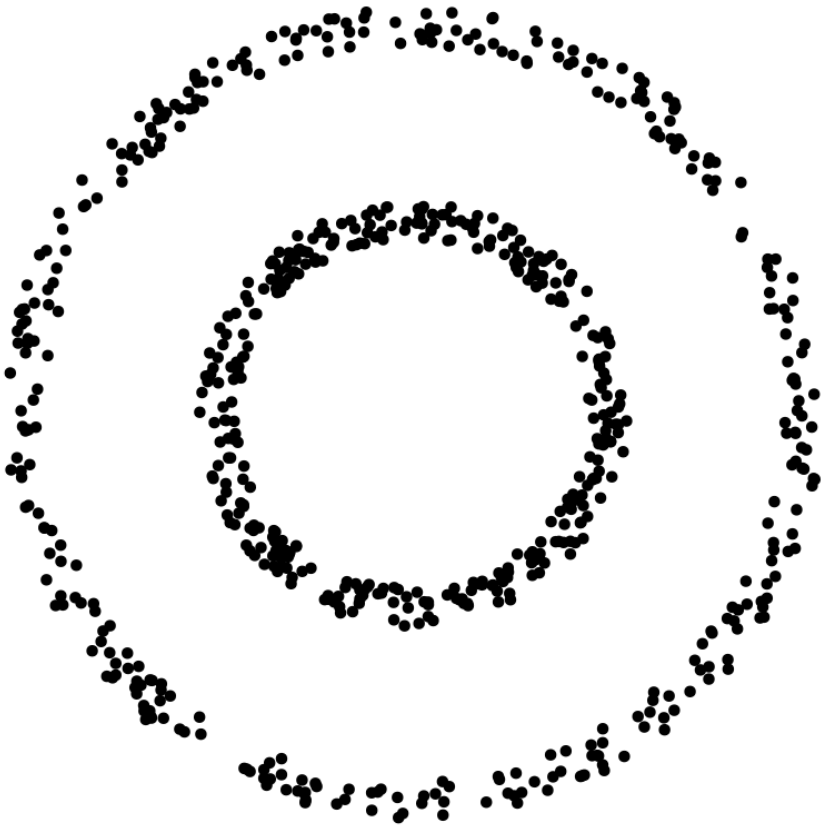


K-means clustering: Optimization

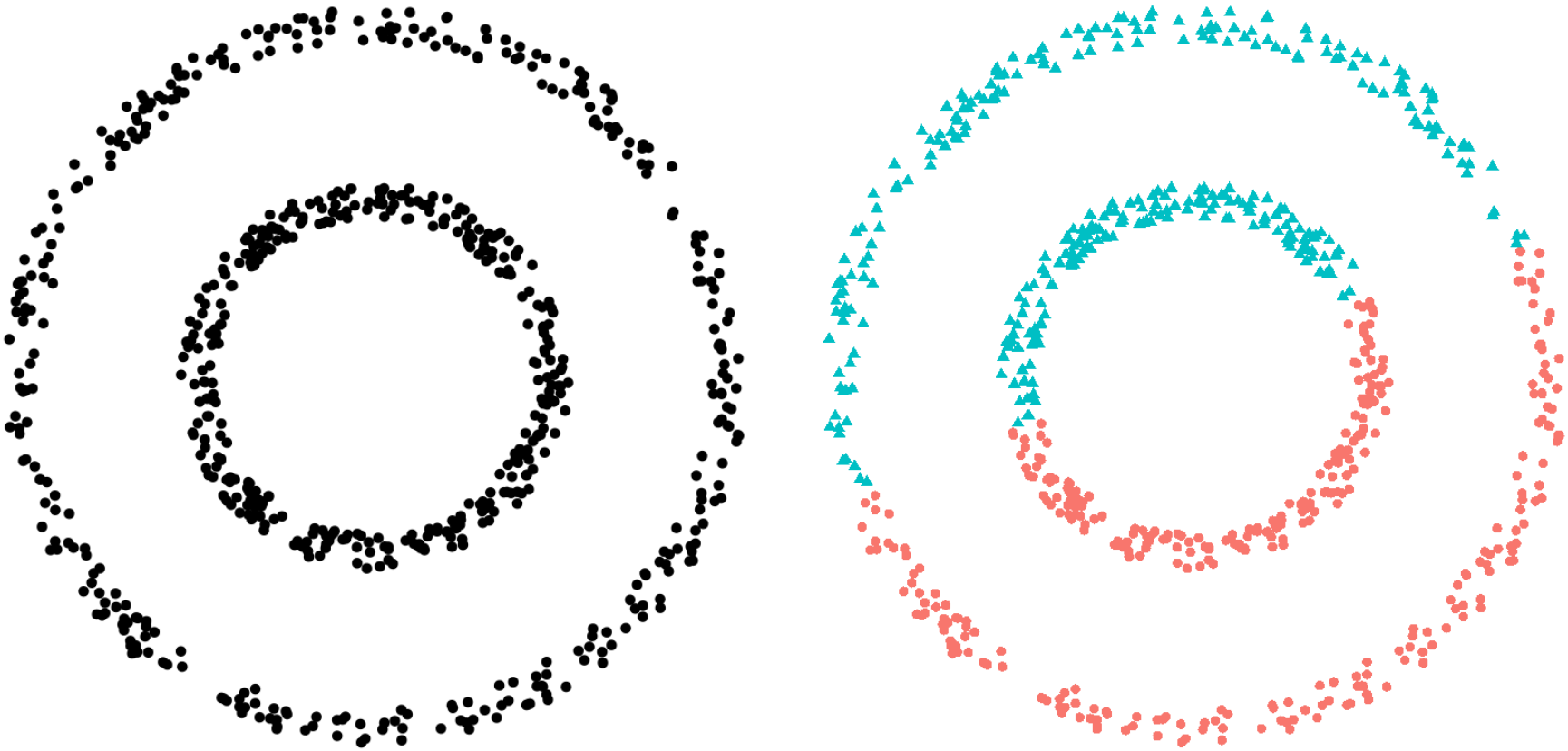


K-means clustering: Local optima

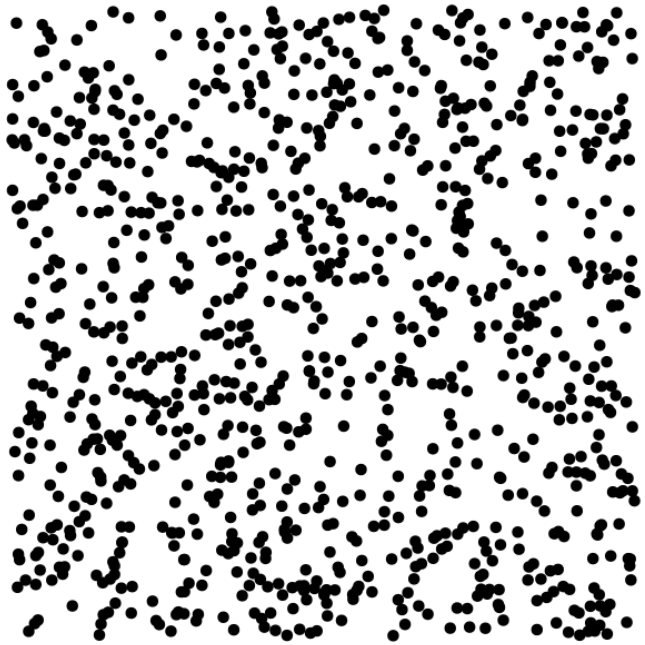
Non-sperical clusters



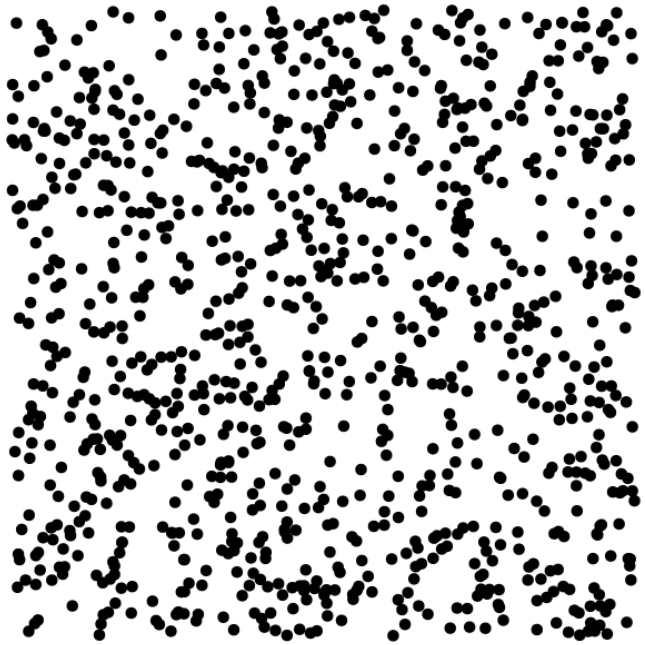
Non-sperical clusters

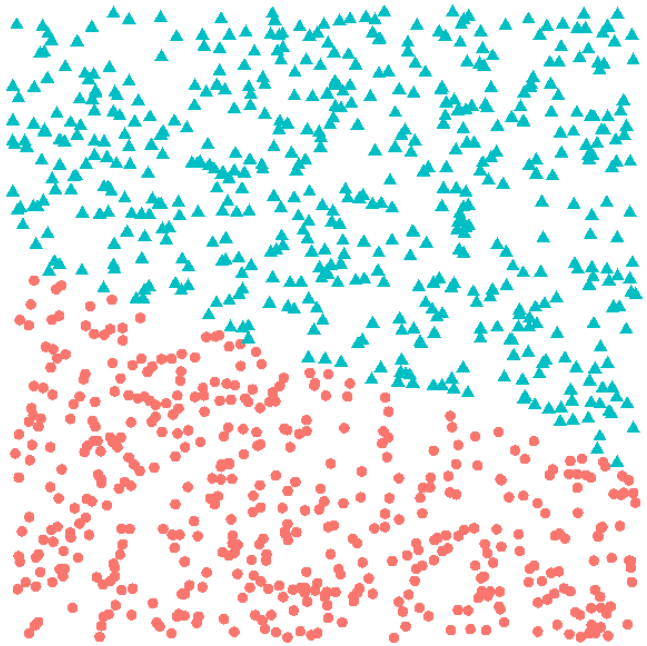


Unequally sized clusters



Unequally sized clusters





K-means clustering: Drawbacks

- Fails if data is not clustered. No guidance how to choose K, and results heavily depend on it.
- Fails if clusters are non-spherical (not spacially grouped).
- Fails if clusters are very different in size.
- Fails if data is binary or categorical (euclidean distance between binary variables).
- Transformations or other clustering approaches (e.g. gaussian mixture models) are possible workarounds.

K-means clustering: Drawbacks

- Local optimum arbitrarily worse than optimal solution.
- Local optimum depends on initial cluster assignment.
- Re-run several times from different random initial assignments.
- Curse of dimensionality: more variation in local optima with larger number of clusters and many features.
- K-means less effective at distinguishing between examples in high dimensions.

➥ Problematic with n-gram frequency representations of text.

K-means clustering with text data

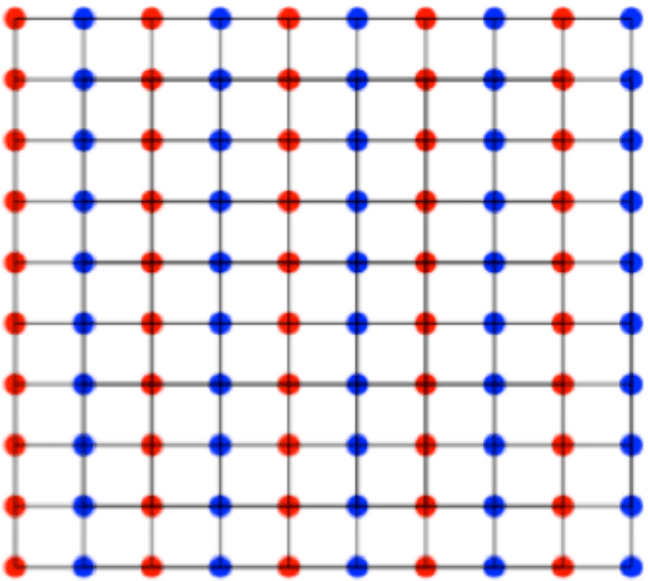
- In general, K-means clustering does not work very well with the large and sparse inputs in text data.
- May be helpful for some exploratory analysis.
- Apply only after substantially reducing the number of features and possibly weighting (e.g. tf-idf). Do not use with binary token indicators.
- Best applied after dimension reduction (e.g. using PCA or a distributional language model).
- Apply if you have meaningful prior information about the number of classes in the data and how they relate to the features, allowing you to make sense of the results.

Digression: Curse of dimensionality

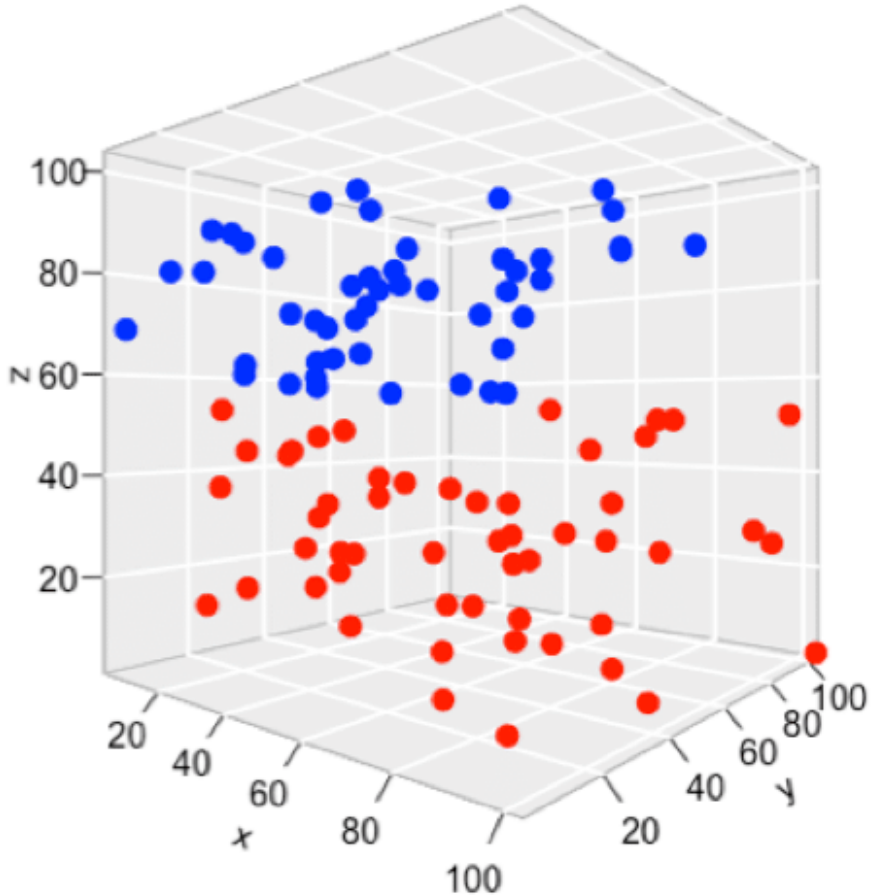
- Nonparametric methods break down in high-dimensional space.
- As dimensions increase, data points become more distant in space.
- The amount of data required to retain the same data density in space increases exponentially with the number of dimensions.
- Classical example for this is the k-nearest-neighbors classifier.
- As the number of dimensions increases, the ratio between the closest distance and the average distance between data points grows. The nearest point will be almost as far away as the closest point.
- This is why K-means does a bad job distinguishing between classes in high dimensions.

Digression: Curse of dimensionality

(A) 1-D

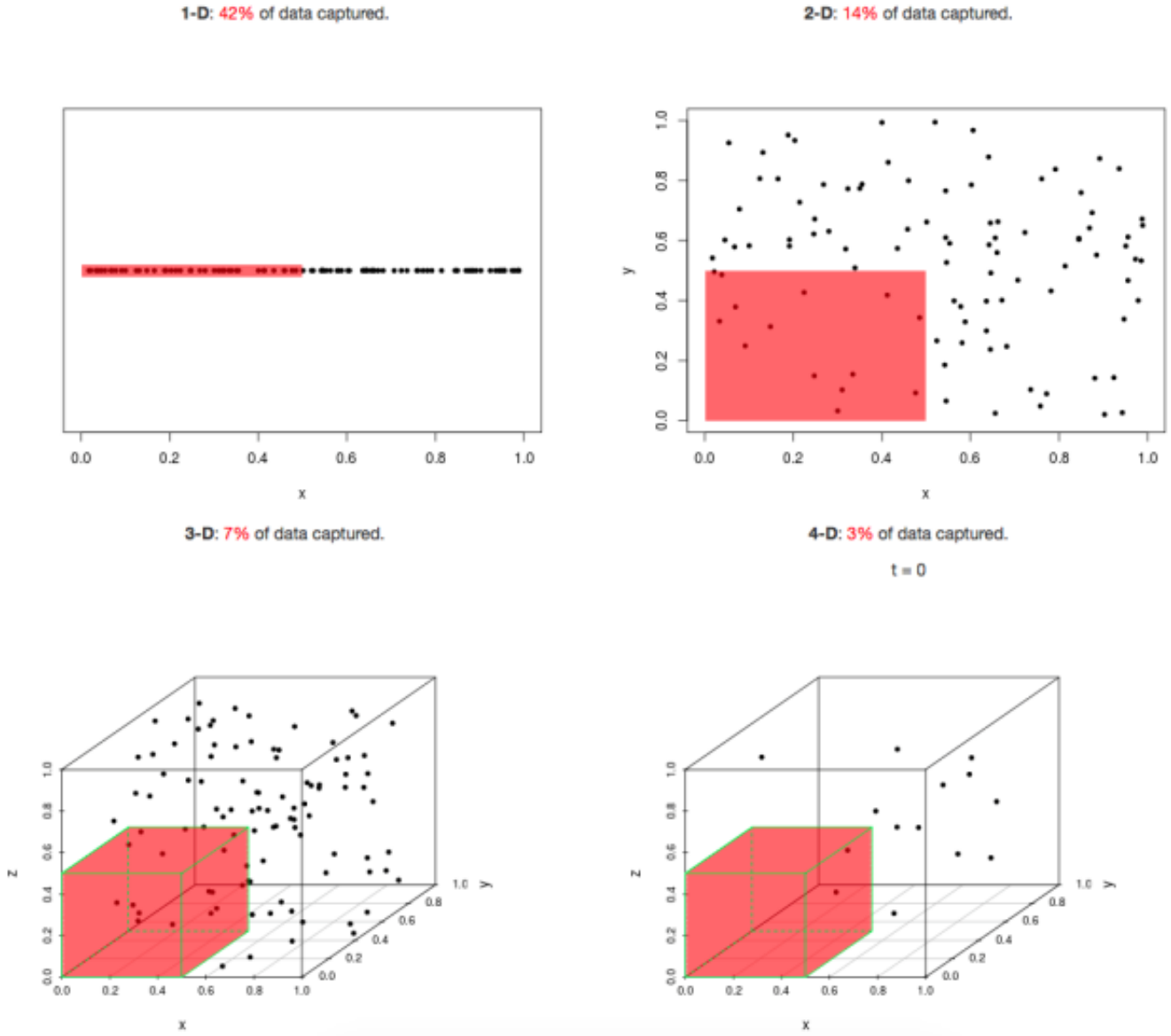


(B) 2-D

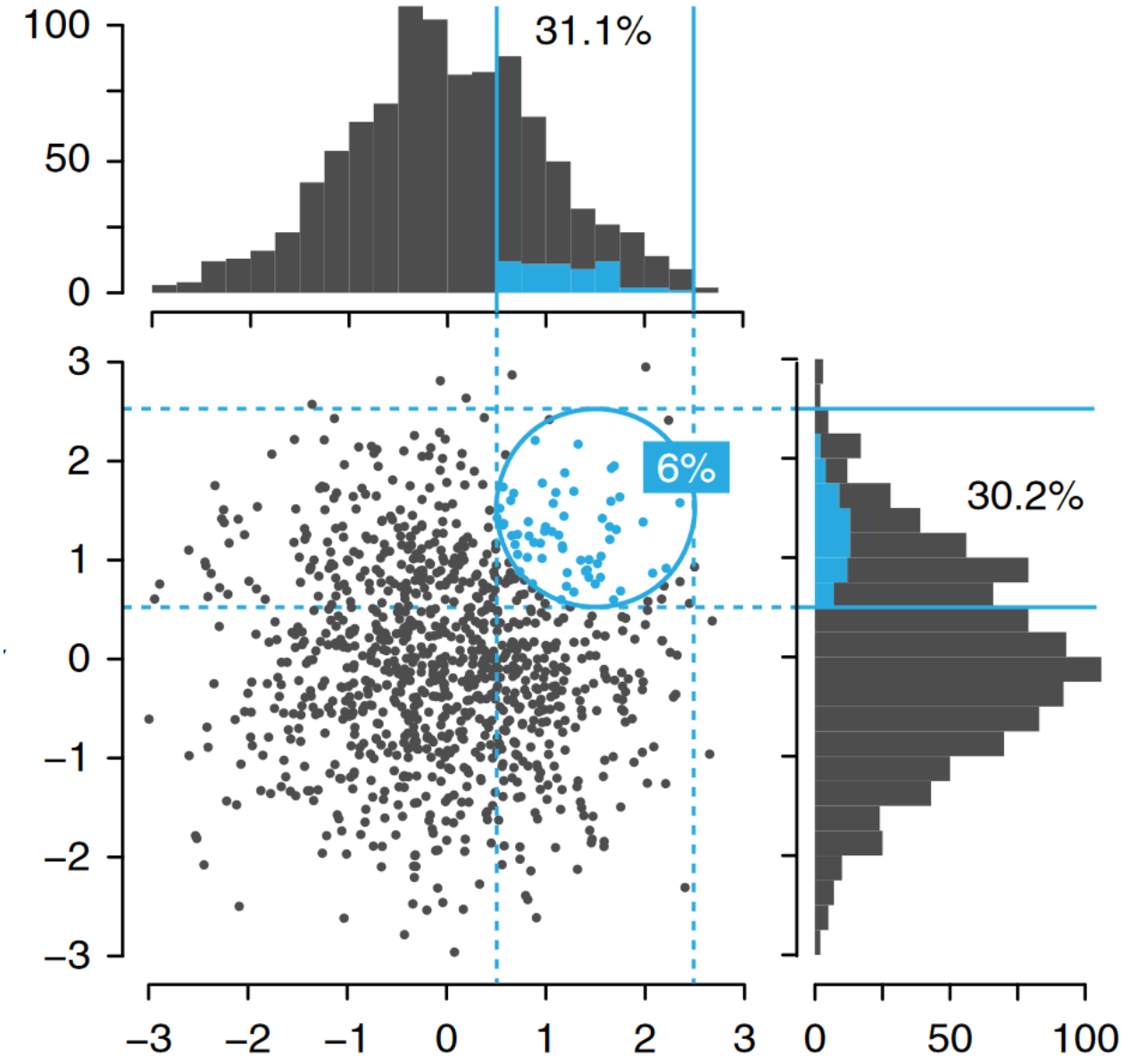


(C) 3-D

Digression: Curse of dimensionality



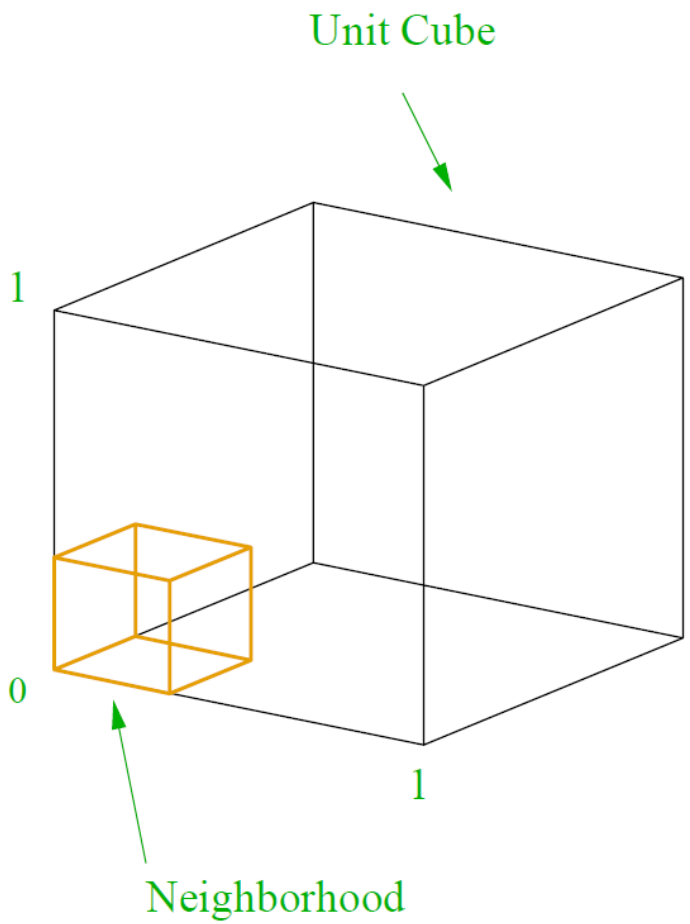
Digression: Curse of dimensionality

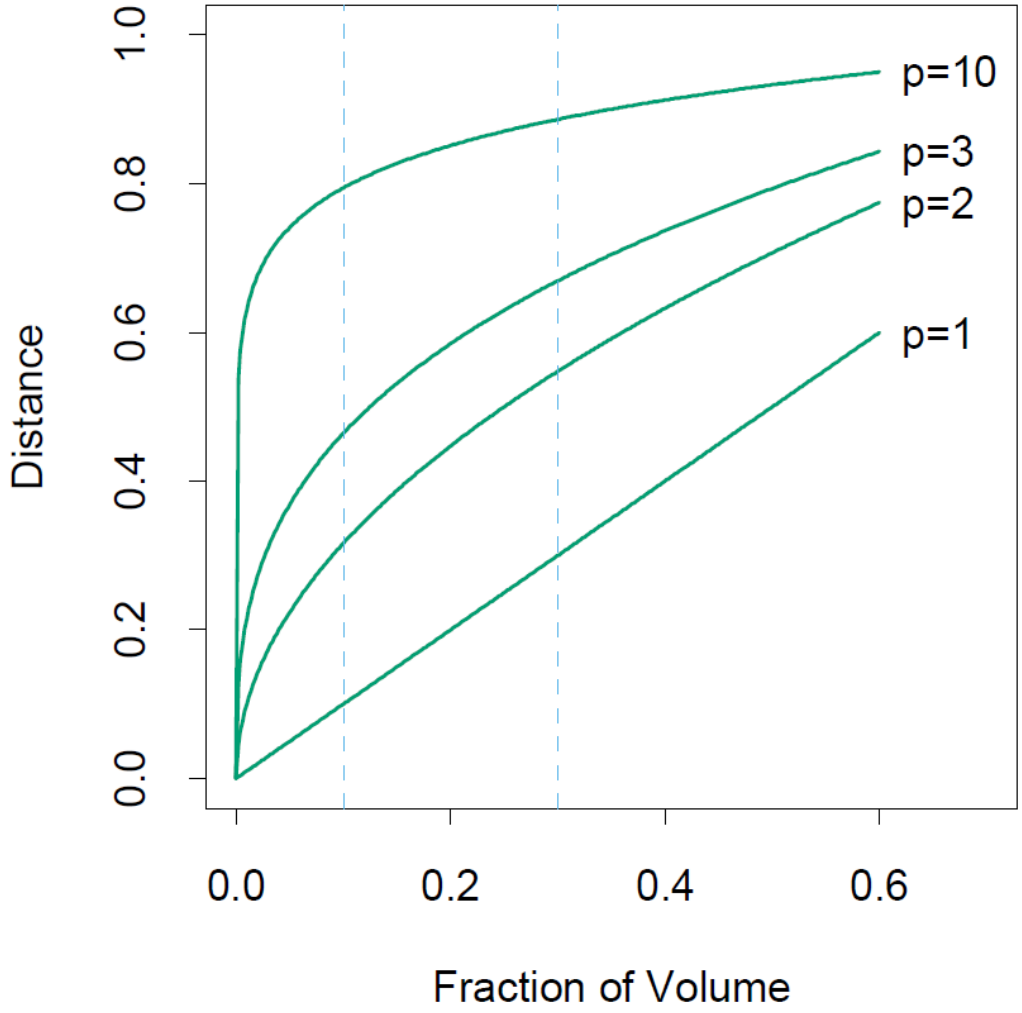


Digression: Curse of dimensionality with KNN

- Assume 3000 points uniformly distributed in a unit hypercube and we want to apply a 3-nearest-neighbor classifier.
- In one dimension, the average distance to capture 3 nearest neighbors is 3/3000 = 0.001.
- In two dimensions, the average distance to capture 3 nearest neighbors is $0.001^{1/2} \approx 0.03$.
- In three dimensions, the average distance to capture 3 nearest neighbors is $0.001^{1/3} = 0.1$.
- In ten dimensions, the average distance to capture 3 nearest neighbors is $0.001^{1/10} \approx 0.5$.

Digression: Curse of dimensionality





Statistical methods

Naive Bayes

Generative language models

- Text regression uses counts as high-dimensional input variables, without attempting to model structure specific to language data.
- Generative models try to learn how attributes influence word choice and account for dependencies between words and attributes.

➥ Define a *generative* model for text tokens $p(c_i|v_i)$.

- Intuitive: Causal relationship typically runs from outcomes to language. Google searches about the flu do not cause the flu.
- Both supervised and unsupervised generative models exist.

Supervised generative models: Naive Bayes

- Most common supervised generative model: Naive Bayes classifier.
- Probabilistic classifier based on applying Bayes' theorem.
- Popular method for text categorization. Good baseline, relatively competitive.
- Relies on a strong naive indepencence assumption of features.
- Assume v is a class to be estimated, a univariate categorical attribute (e.g. spam). We are interested in the probability that a document d is a member of class v.

➥ How likely is it that a document is spam?

- Probabilistic classifier: $\hat{v} = \underset{v \in V}{\operatorname{argmax}}\, p(v|d)$

Naive Bayes

- Rewrite class estimate using Bayes' rule:

$$\hat{v} = \underset{v \in V}{\operatorname{argmax}}\, p(v|d) = \underset{v \in V}{\operatorname{argmax}}\, \frac{p(d|v)p(v)}{p(d)} \,.$$

- Drop the denominator. We compute the class probability for each possible class, but $p(d)$ does not change for each class.
- Choose the class that maximizes

$$\hat{v} = \underset{v \in V}{\operatorname{argmax}}\, p(v|d) = \underset{v \in V}{\operatorname{argmax}}\, p(d|v)p(v) \,.$$

- Most probable class $\hat{v}$ has the highest product of two probabilities: the *prior probability* of the class $p(v)$, and the *likelihood* of the document $p(d|v)$.

Naive Bayes

- Without loss of generalization, we can represent a document d as a set of features $f_1, f_2, \ldots, f_p$:

$$\hat{v} = \underset{v \in V}{\operatorname{argmax}}\, p(f_1, f_2, \ldots, f_p|v)p(v) \; .$$

- Too difficult to compute directly. Make two simplifying assumptions.

1. Bag-of-words: Position does not matter, features $f_1, f_2, \ldots, f_p$ only encode word identity and not position (as previously).
2. Naive Bayes/conditional independence assumption: The probabilities $p(f_i|v)$ are independent given class v and can be 'naively' multiplied:

$$p(f_1, f_2, \ldots, f_p|v) = p(f_1|v) \cdot p(f_2|v) \cdot \ldots \cdot p(f_p|v)$$

Naive Bayes

- Class chosen is thus

$$\hat{v} = \underset{v \in V}{\operatorname{argmax}}\, p(v) \prod_{f \in F} p(f|v) \,.$$

- Considering only word tokens:

$$\hat{v} = \underset{v \in V}{\operatorname{argmax}}\, p(v) \prod_{c \in \mathcal{V}} p(c|v) \,.$$

- In log space to avoid underflow and increase speed:

$$\hat{v} = \underset{v \in V}{\operatorname{argmax}}\, \log p(v) + \sum_{c \in \mathcal{V}} \log p(c_j|v) \,.$$

Naive Bayes classifier: Remarks

- How can we learn $p(v)$ and $p(f_j|v)$?
- Consider using the frequencies in the data.
- Learn $p(v)$ using the share of documents in class v in the training data.

$$\hat{p}(v) = \frac{N_c}{N_{doc}}$$

- Learn $p(c_j|v)$ as the fraction of times the word c_j appears among all words in the vocabulary in all documents of class v.

$$\hat{p}(c_j|v) = \frac{count(c_j, v)}{\sum_{c \in \mathcal{V}} count(c, v)}$$

Training the Naive Bayes classifier

- Some words may not appear in a given class in the training data, e.g.

$$\hat{p}(\text{'fantastic'}|\text{spam}) = \frac{count(\text{'fantastic'}, \text{spam})}{\sum_{j=1}^{p} count(c_j, \text{spam})} = 0 \ .$$

- Naive bayes multiplies all the feature likelihoods together.
- Zero proabilitities in the likelihood term for any class will cause the probability of the class to be zero.
- Simplest solution: Laplace (add-one) smoothing.

$$\hat{p}(c_j|v) = \frac{count(c_j, v) + 1}{\sum_{c \in \mathcal{V}} count(c, v) + 1} = \frac{count(c_j, v) + 1}{\left(\sum_{c \in \mathcal{V}} count(c, v)\right) + |\mathcal{V}|}$$

Training the Naive Bayes classifier

- Ignore new words in test data that do not occur in training sample.
- Independence assumption rules out the possibility that using one token (‘hello’) influences the probability of using another (‘hi’).
- Even though this may be unrealistic, naive Bayes works suprisingly well in practice. Often used in spam filters.
- Removing stop words does not necessarily improve performance.

Statistical methods

Topic model

Unsupervised generative models

- No direct observations of the true attributes v_i.
- Inference about attributes depends entirely on assumptions imposed on the structure of the model $p(\mathrm{c}_i|\mathrm{v}_i)$.
- Typical model: Each observation c_i is a conditionally independent draw from the vocabulary of possible tokens according to some document-specific token probability vector $\mathrm{q}_i = [q_{i1} \dots q_{ip}]'$.
- Conditioning on document length $m_i = \sum_j c_{ij}$, this implies a multinomial distribution for the counts.

$$\mathrm{c}_i \sim \mathrm{MN}(\mathrm{q}_i, m_i)$$

- This multinomial model underlies many contemporary statistical models for text. Popular variant: Topic model (Blei et al. 2003, Blei 2012).

Topic models

- Most popular topic model is Latent Dirichlet Allocation (LDA).
- The idea is to find latent themes (*topics*) in documents.
- Essentially a clustering problem. Think of both words and documents being clustered.
- Basic idea: Every document is a mixture of topics.
  Every topic is a mixture of words.

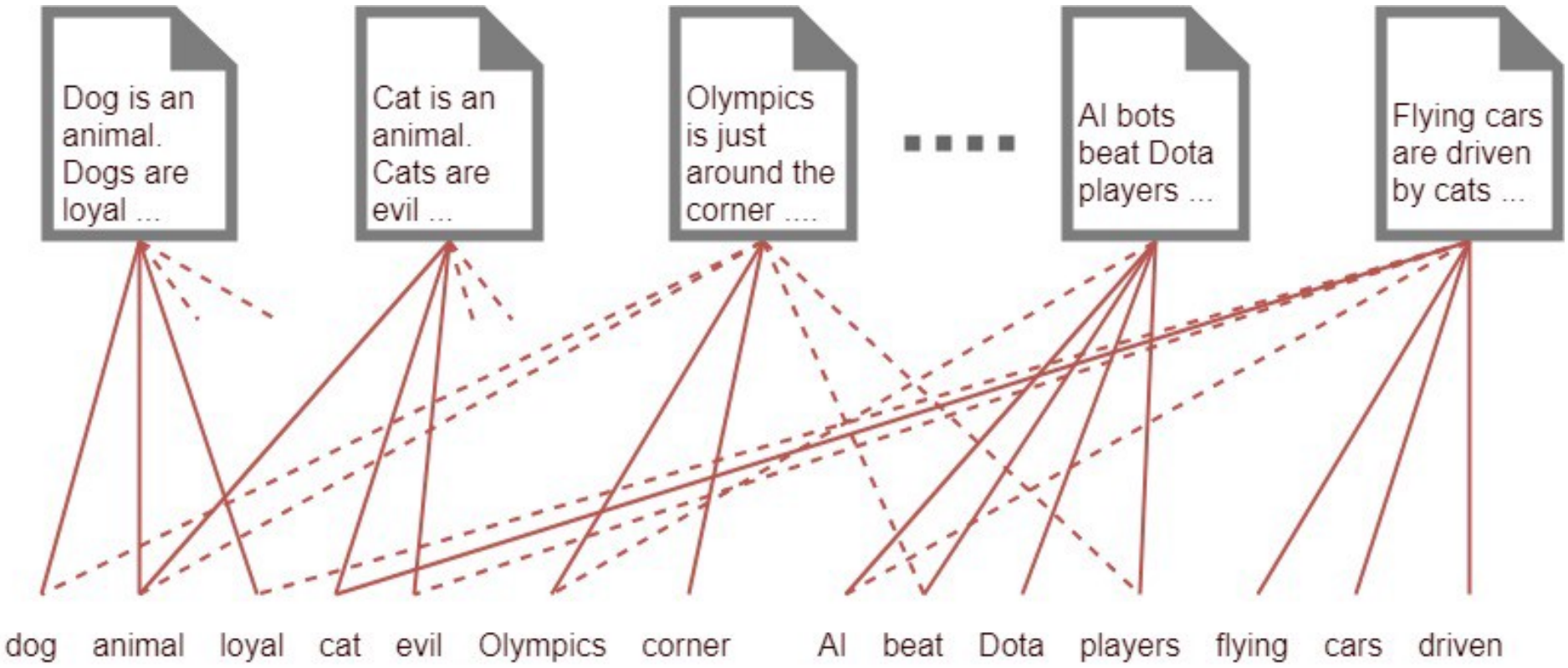
Topic models

- Topic models can be understood as factor models for the normalized token counts.
- $\mathrm{c}_i \sim \mathrm{MN}(\mathrm{q}_i, m_i)$, link function $\mathrm{q}_i = q(\mathrm{v}_i)$ links text and attributes.
- *Topic model* specification:

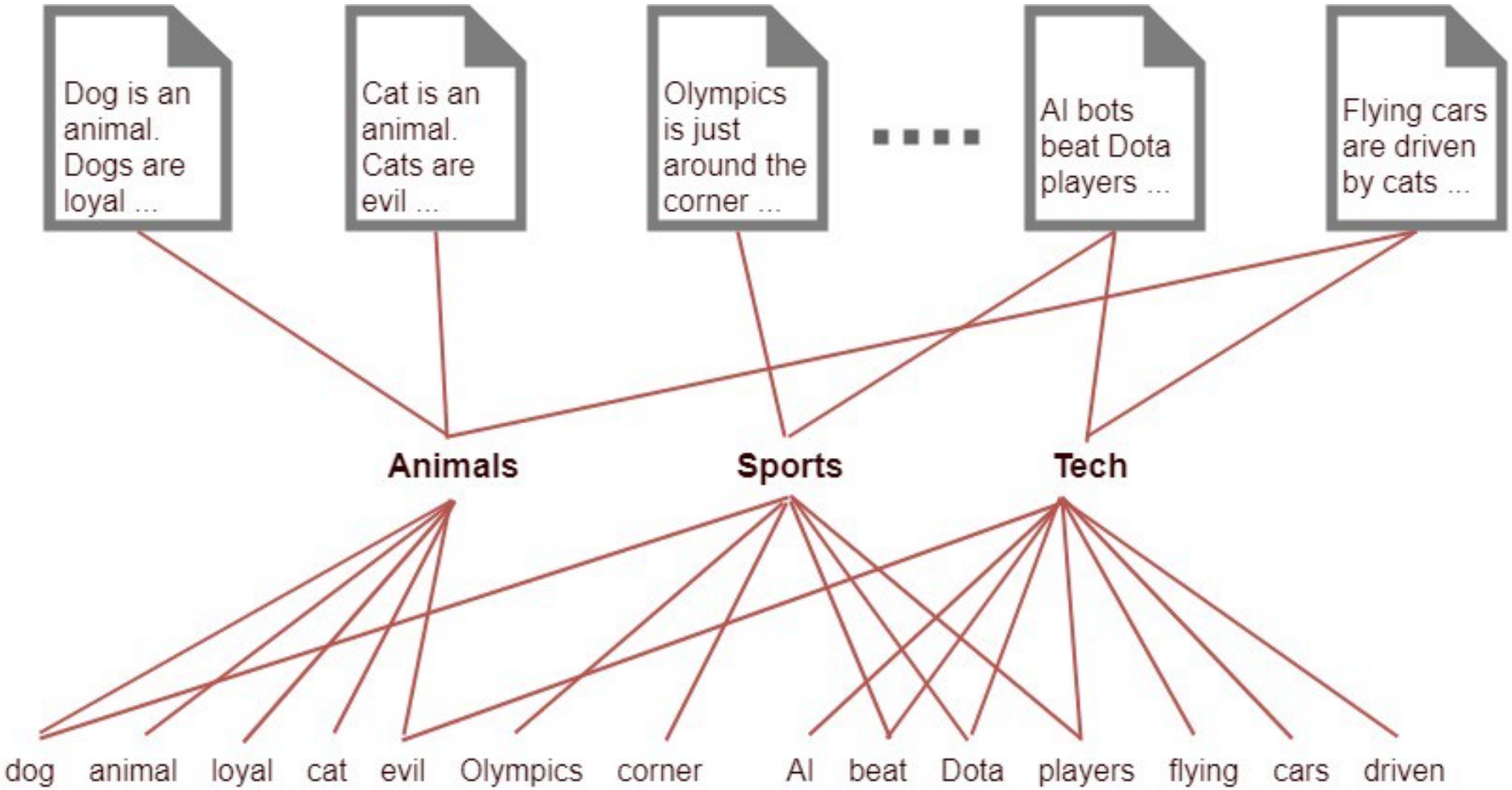
$$E\left(\frac{\mathrm{c}_i}{m_i}\right) = \mathrm{q}_i = v_{i1}\boldsymbol{\theta}_1 + v_{i2}\boldsymbol{\theta}_2 + \ldots + v_{ik}\boldsymbol{\theta}_k = \boldsymbol{\Theta}\mathrm{v}_i$$

- Each topic is a probability vector over possible tokens, $\boldsymbol{\theta}_l$, $l = 1, \ldots, k$ where $\theta_{lj} \geq 0$ and $\sum_{j=1}^{p} \theta_{lj} = 1$
- Latent attributes v_{il} are referred to as *topic weights* and restricted such that $v_{il} \geq 0$ and $\sum_{l=1}^{k} v_{il} = 1$.

Topic model


Dog is an animal. Dogs are loyal ...
Cat is an animal. Cats are evil ...
Olympics is just around the corner ....
AI bots beat Dota players ...
Flying cars are driven by cats ...
dog
animal
loyal
cat
evil
Olympics
corner
AI
beat
Dota
players
flying
cars
driven

Topic model


Dog is an animal. Dogs are loyal ...
Cat is an animal. Cats are evil ...
Olympics is just around the corner ...
AI bots beat Dota players ...
Flying cars are driven by cats ...
Animals
Sports
Tech
dog
animal
loyal
cat
evil
Olympics
corner
AI
beat
Dota
players
flying
cars
driven

Topic model/LDA

- Suppose your corpus consists of the following sentences.

```
I like to eat broccoli and bananas.
I ate a banana and spinach smoothie for breakfast.
Chinchillas and kittens are cute.
My sister adopted a kitten yesterday.
Look at this cute hamster munching on a piece of broccoli.
```

- Given this data and asked for two topics, we might find something like:

```
Sentences 1 and 2: 100% Topic A
Sentences 3 and 4: 100% Topic B
Sentence 5:        60% Topic A, 40% Topic B

Topic A: 30% broccoli, 15% bananas, 10% breakfast, 10% munching, ...
Topic B: 20% chinchillas, 20% kittens, 20% cute, 15% hamster, ...
```

➥ Interpret interpret topic A to be about food, topic B to be about animals.

LDA topic todel: Key features

- Documents exhibit multiple topics (typically relatively few).
- Probabilistic model with a corresponding generative process (each document generated by this process).
- A topic is a distribution over a fixed vocabulary. Topics are assumed to be generated first.
- Only the number of topics is specified in advance.

LDA topic todel: Generative process

To generate a document:

1. Choose a distribution over topics, i.e. a topic mixture for the document.
2. Generate each word w_i in the document by:
   - 2.1 Pick a topic from the distribution over topics.
   - 2.2 Pick a word from the corresponding topic (distribution over vocabulary).

LDA topic todel: Generative process

To generate a document:

1. Choose a distribution over topics, i.e. a topic mixture for the document.
   (Dirichlet distribution over a fixed set of k topics.)
2. Generate each word w_i in the document by:
   2.1 Pick a topic from the distribution over topics.
   (From the multinomial distribution sampled above.)
   2.2 Pick a word from the corresponding topic (distribution over vocabulary).
   (From the topic's multinomial distribution.)

- Step 1 requires a distribution over a distribution.
- Words are generated independently of other words (unigram bag-of-words model).
- Assuming this process, LDA tries to backtrack from the documents to find a set of topics that are likely to have the generated the collection.

LDA topic model: Generative process

- Notation:
  - $\boldsymbol{\beta}_{1:K}$ are the topics, each $\boldsymbol{\beta}_k$ is a distribution over the token vocabulary.
  - $\boldsymbol{\theta}_d$ are the weights/proportions for document d.
  - $\theta_{d,k}$ are the weights/proportions for topic k in document d.
  - z_d are the topic assignments for document d.
  - $z_{d,n}$ are the topic assignments for word n in document d.
  - w_d are the observed words for document d.
- The joint distribution of hidden and observed variables:

$$p(\boldsymbol{\beta}_{1:K}, \boldsymbol{\theta}_{1:D}, z_{1:D}, w_{1:D}) = \prod_{k=1}^{K} p(\boldsymbol{\beta}_k) \prod_{d=1}^{D} p(\boldsymbol{\theta}_d) \prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\boldsymbol{\beta}_{1:K}, z_{d,n})$$

LDA topic model: Estimation

- v_i are latent, use alternating inference for $V|\Theta$ and $\Theta|V$.
- Use the expectation maximization algorithm to maximize the implied likelihood (or an equivalent Bayesian specification).
- Number of topics k is the tuning parameter.
- Choice of k often arbitrary. Daten-driven choice methods exist. Start with a sensible number, adjust to improve interpretability.
- Beware the pitfalls of local optima.

Topic models

- Topic model hugely popular since its introduction (Blei et al. 2003, Grimmer and Stewart 2013).
- Output from topic models can be used as input for further analysis.
- Researchers in political science attach political issues and beliefs to the latent topics (e.g. Monroe et al. 2008).
- Analysis of open-ended survey response (Roberts et al. 2014).

Expectation maximization algorithm

- Given a known distribution, the efficient way to estimate unknown parameters is maximum likelihood (estimate parameters by maximizing the likelihood that the data was produced by a given model).
- But: We do not observe all data, the dependent variable (topic class) is latent.
- *Both* parameters and part of the data are unknown. Cannot take the derivative of the likelihood and solve.
- Marginal likelihood $\ell(\theta, C) = p(C|\theta) = \int p(C, V|\theta) dV$ is intractable.
- Solution to ML is a set of interlocking equations: Solving for the parameters requires the values of the latent variables and vice versa.

➥ Solution: Expectation Maximization algorithm (Dempster et al. 1977). Enables estimation of unknown parameters even though data is missing.

EM algorithm: Basic idea

- EM algorithm provides a way to obtain estimates for the latent classes.
- Begin from the basic idea that the two sets of equations can be solved numerically.
- Pick arbitraty parameter values, use them to estimate the latent class values, then use these to get a better estimate of the parameters.
- Keep alternating until both converge to fixed points.
- Will converge to a maximum or a saddle point of the likelihood.
- No guarantee for a global maximum. Repeat multiple times with different starting values.

EM algorithm

1. Initialize random parameter values.
2. **E-step**

   Compute latent attributes V from current parameter estimates θ and feature data C. At the j-th iteration, compute

   $$Q(\theta|\hat{\theta}^{(j+1)}) = \mathbb{E}[\ell(\theta, V, C|C, \hat{\theta}^{(j)})] \, .$$

3. **M-step**

   Estimate parameters θ using class membership and feature data (V, C) by applying maximum likelihood. Determine the new estimate $\hat{\theta}^{(j+1)}$ as

   $$\hat{\theta}^{(j+1)} = \arg\max_{\theta} Q(\theta|\hat{\theta}^{(j)}) \, .$$

4. Iterate 2. and 3. until convergence.

Outlook: 'Deep learning' and word embedding

- Many other machine learning methods applied to text.
- Among the most common are neural networks.
- Classical neural nets tended to overfit and be difficult to tune in high-dimensional noisy settings (like text analysis).
- Recently popular 'deep' versions (many layers, fewer nodes per layer) work better, faster and require less tuning even in difficult problems.
- State-of-the-art, these power most of the text functions on your phone and computer, e.g. translation services or syntactic parsing.
- Often rely on word embedding representations (next lecture).
- Structure varies greatly across applications.

Applications

Inferring autorship: Federalist papers

Skip

- First modern statistical analysis of text data, Mosteller and Wallace (1963): Who wrote the contested Federalist Papers?
- Federalist papers were 77 essays written by Alexander Hamilton, John Jay and James Madison.
- Published in newspapers in 1787–1788 to persuade citizens of New York State to ratify the constitution.
- Authorship of 12 papers is contested between Hamilton and Madison.

Inferring autorship: Federalist papers

Skip

- Determining authorship mostly relies on the frequency of function words (stop words).
- Use naive Bayes trained on Documents known to be written by either Hamilton or Madison to predict the author of the contested papers.
- Features are the unigram frequencies of function words.
- Majority of papers were most likely written by Madison.

Inferring autorship: IV

- Stock and Trebbi (2003): Who invented instrumental variables?
- ➥ Was the mathematical appendix to Wright (1928) 'The Tariff on Animal and Vegetable Oils' written by his son?
- Data features counts of function words and counts of certain grammatical constructions $c_i = (c_i^{func}, c_i^{gram})$.
- Training sample consists of 45 documents known to be written by either author.
- Test sample consists of eight blocks from the appendix and one from the main book for validation.
- Authors extract the first four principal components from $c_i = (c_i^{func}, c_i^{gram})$, then regress them on the binary authorship obtaining $(\hat{v}_i^{func}, \hat{v}_i^{gram})$.

Who invented IV?

Skip

Summary Statistics for the Six Stylometric Indicators with the Largest *t*-Statistics

	Philip		*Sewall*			*Appendix B*	
	Mean	*Standard Deviation*	*Mean*	*Standard Deviation*	*t*	*Mean*	*Standard Deviation*
noun followed by coordinating conjunction	26.8	7.0	17.3	4.6	5.55	27.0	5.0
to	29.5	5.8	20.9	6.1	4.79	28.0	8.6
now	1.6	1.5	0.1	0.3	4.74	1.1	1.0
when	2.4	2.1	0.3	0.7	4.72	1.8	1.2
in	22.7	5.3	29.8	5.5	−4.34	18.5	5.8
so	2.1	1.6	0.7	0.8	3.82	2.0	1.7
n	25		20			6	

Notes: The entries in columns 2 and 3 are the mean and standard deviations of the counts per 1,000 words of the stylometric indicator in column 1 in the 25 blocks undisputedly written by Philip Wright. Columns 4 and 5 contain this information for the 20 blocks undisputedly written by Sewall Wright. The next column contains the two-sample *t*-statistic testing the hypothesis that the mean counts are the same for the two authors. The final two columns contain means and standard deviations for the 6 blocks from Appendix B. Shaded indicators occur in the excerpt in Exhibit 2.

Who invented IV?

Skip

Cross-Validation Estimates of Accuracy Rates of Assigned Authorship

	Principal Components Regression		*Linear Discriminant Analysis*	
	Predicted Author:		*Predicted Author:*	
True Author:	*Sewall*	*Philip*	*Sewall*	*Philip*
Sewall	100%	0%	90%	10%
Philip	0%	100%	0%	100%

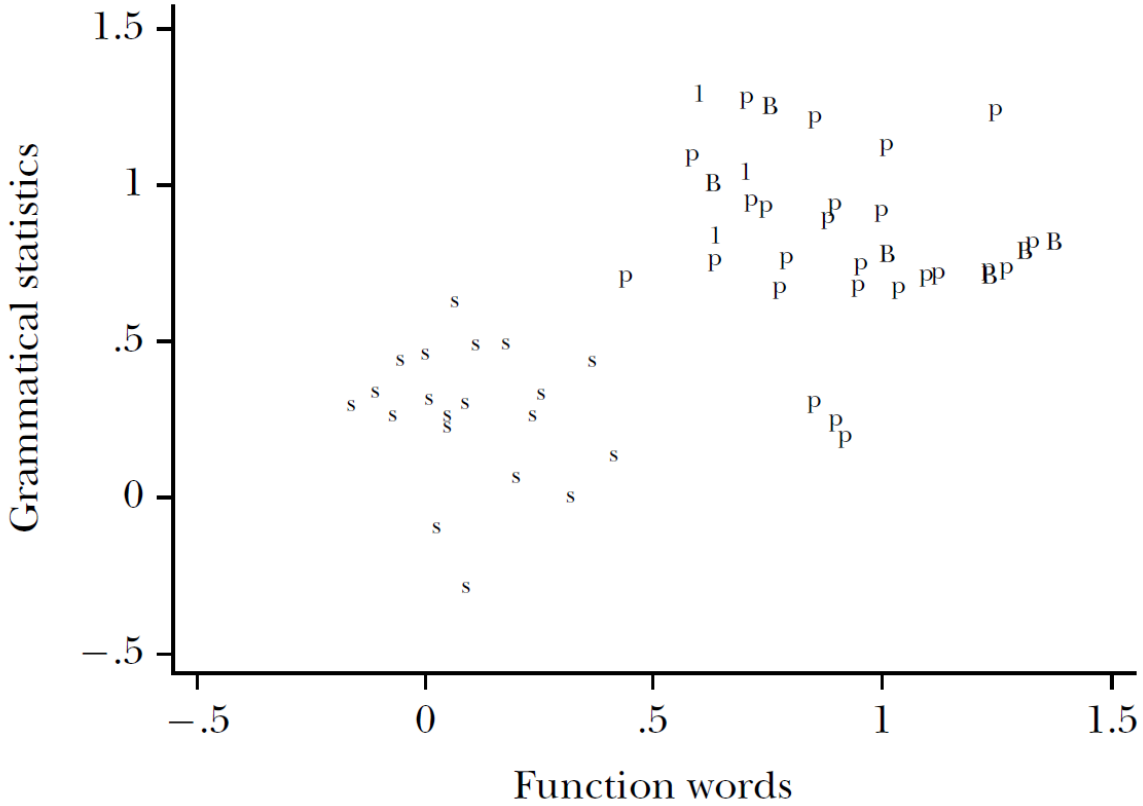
Notes: Based on leave-one-out cross-validation analysis of 45 1,000-word blocks of known authorship.

Who invented IV?

Skip

Scatterplot of Predicted Values from Regression on First Four Principal Components: Grammatical Statistics versus Function Words

s = block undisputedly written by Sewall Wright
p = block undisputedly written by Philip G. Wright
1 = block from chapter 1, *The Tariff on Animal and Vegetable Oils*
B = block from Appendix B, *The Tariff on Animal and Vegetable Oils*



Stock prices

Skip

- Dictionary approach: Cowles 1933 categorized Wall Street Journal editorial articles as ‘bullish’, ‘bearish’ or ‘doubtful’ and used classifications to predict future Dow Jones returns.
- Many other modern applications using media announcements, Twitter messages, WSJ or investor publications.
- Mostly dictionary approaches, some text regression, few generative models.
- Related literature on central bank sentiment and announcements.

Nowcasting

Skip

- Indicators like unemployment, retail sales and GDP are measured at low frequency and estimates are released with substantial lag.
- Text produced online can be used to construct alternative real-time estimates of the current values of these variables.
- Google Flu Trends project (Ginsberg et al. 2009, Butler 2013, Lazer et al. 2014).
- Nowcasting macroeconomic variables (Choi and Varian 2012, Scott and Varian 2014, Scott and Varian 2015, Varian 2014).

Proxying for unobservable measures

Skip

- Other variables like local government corruption or racial prejudice are not easily measured at all.
- Text can often be used to construct measures of variables which are otherwise difficult to capture or survey.
- Racial prejudice and voting (Stephens-Davidowitz 2014).
- Corruption in US cities (Saiz and Simonsohn 2013).

> *"The internet contains billions of documents. We show that document frequencies in large decentralized textual databases can capture the cross-sectional variation in the occurrence frequencies of social phenomena. We characterize the econometric conditions under which such proxying is likely. [...]"*

Saiz and Simonsohn 2013
Proxying for Unobservable Variables with Internet Document-Frequency

Media Slant

- Groseclose and Milyo (2005) identify political slant of media outlets.
- Develop a model using speeches by politicians in the US congress and a left-right political ideology score from Americans for Democratic Action.
- What kind of politician does a news outlet's content sound most familiar to?
- Drastic dimension reduction by only looking at the occurence of a (supposedly) informative subset, the names of 200 think tanks.
- Gentzkow and Shapiro (2010) do a similar analysis but omit the think tank restriction.

Media Slant

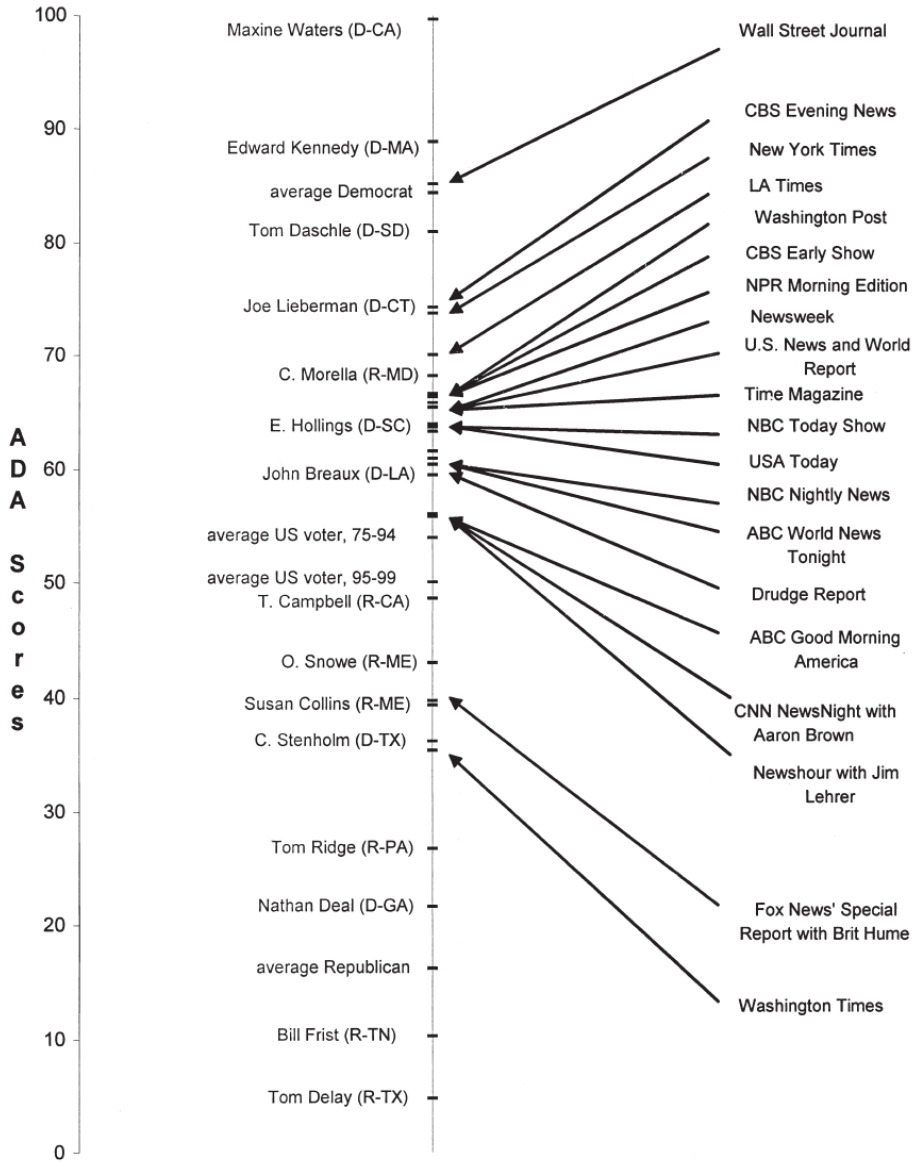


FIGURE II
Adjusted ADA Scores of Selected Politicians and Media Outlets

TABLE I

MOST PARTISAN PHRASES FROM THE 2005 *CONGRESSIONAL RECORD*[a]

Panel A: Phrases Used More Often by Democrats

Two-Word Phrases

private accounts	Rosa Parks	workers rights
trade agreement	President budget	poor people
American people	Republican party	Republican leader
tax breaks	change the rules	Arctic refuge
trade deficit	minimum wage	cut funding
oil companies	budget deficit	American workers
credit card	Republican senators	living in poverty
nuclear option	privatization plan	Senate Republicans
war in Iraq	wildlife refuge	fuel efficiency
middle class	card companies	national wildlife

Three-Word Phrases

veterans health care	corporation for public broadcasting	cut health care
congressional black caucus	additional tax cuts	civil rights movement
VA health care	pay for tax cuts	cuts to child support
billion in tax cuts	tax cuts for people	drilling in the Arctic National
credit card companies	oil and gas companies	victims of gun violence
security trust fund	prescription drug bill	solvency of social security
social security trust	caliber sniper rifles	Voting Rights Act
privatize social security	increase in the minimum wage	war in Iraq and Afghanistan
American free trade	system of checks and balances	civil rights protections
central American free	middle class families	credit card debt

TABLE I—*Continued*

Panel B: Phrases Used More Often by Republicans

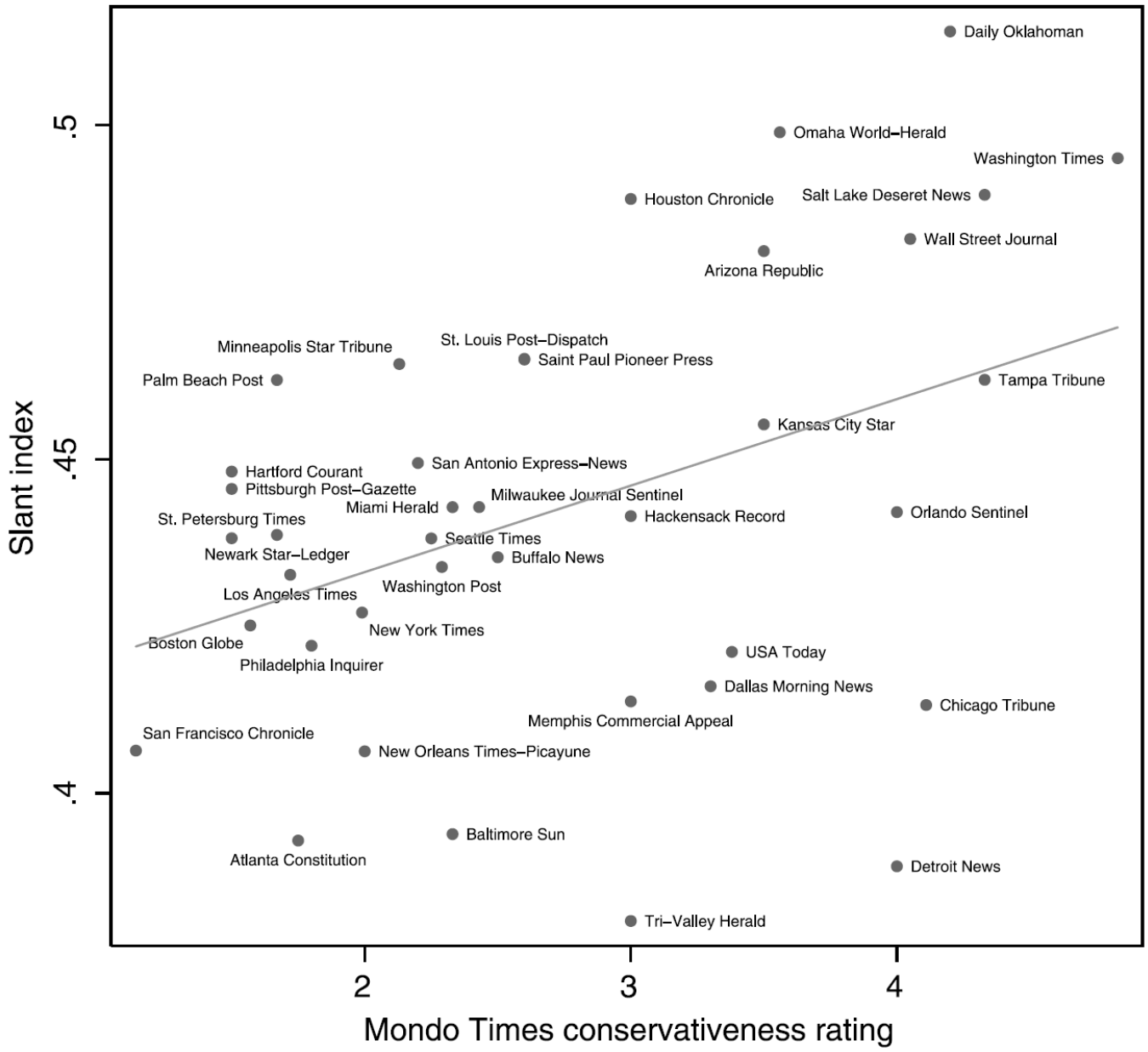
Two-Word Phrases

stem cell	personal accounts	retirement accounts
natural gas	Saddam Hussein	government spending
death tax	pass the bill	national forest
illegal aliens	private property	minority leader
class action	border security	urge support
war on terror	President announces	cell lines
embryonic stem	human life	cord blood
tax relief	Chief Justice	action lawsuits
illegal immigration	human embryos	economic growth
date the time	increase taxes	food program

Three-Word Phrases

embryonic stem cell	Circuit Court of Appeals	Tongass national forest
hate crimes legislation	death tax repeal	pluripotent stem cells
adult stem cells	housing and urban affairs	Supreme Court of Texas
oil for food program	million jobs created	Justice Priscilla Owen
personal retirement accounts	national flood insurance	Justice Janice Rogers
energy and natural resources	oil for food scandal	American Bar Association
global war on terror	private property rights	growth and job creation
hate crimes law	temporary worker program	natural gas natural
change hearts and minds	class action reform	Grand Ole Opry
global war on terrorism	Chief Justice Rehnquist	reform social security

Media Slant


Daily Oklahoman
Omaha World–Herald
Washington Times
Houston Chronicle
Salt Lake Deseret News
Wall Street Journal
Arizona Republic
Minneapolis Star Tribune
St. Louis Post–Dispatch
Saint Paul Pioneer Press
Palm Beach Post
Tampa Tribune
Kansas City Star
Hartford Courant
San Antonio Express–News
Pittsburgh Post–Gazette
Milwaukee Journal Sentinel
Miami Herald
St. Petersburg Times
Hackensack Record
Orlando Sentinel
Seattle Times
Newark Star–Ledger
Buffalo News
Washington Post
Los Angeles Times
New York Times
Boston Globe
Philadelphia Inquirer
USA Today
Dallas Morning News
Chicago Tribune
Memphis Commercial Appeal
San Francisco Chronicle
New Orleans Times–Picayune
Baltimore Sun
Atlanta Constitution
Detroit News
Tri–Valley Herald
Slant index
.5
.45
.4
2
3
4
Mondo Times conservativeness rating

Topics in research and political debate

- Several studies apply topic models to describe how the focus of attention in specific text corpora shifts over time.
- Topics in *Science* (Blei and Lafferty 2007).
- Quinn et al. (2010) use a dynamic topic model to identify issues being discussed in the US Senate and track their relative importance over time. Preferred specification has 42 topics which appear coherent.
- Monroe et al. (2008) analyze words that identify partisanship within the topics identified by Quinn et al. (2010).

Topics in Science

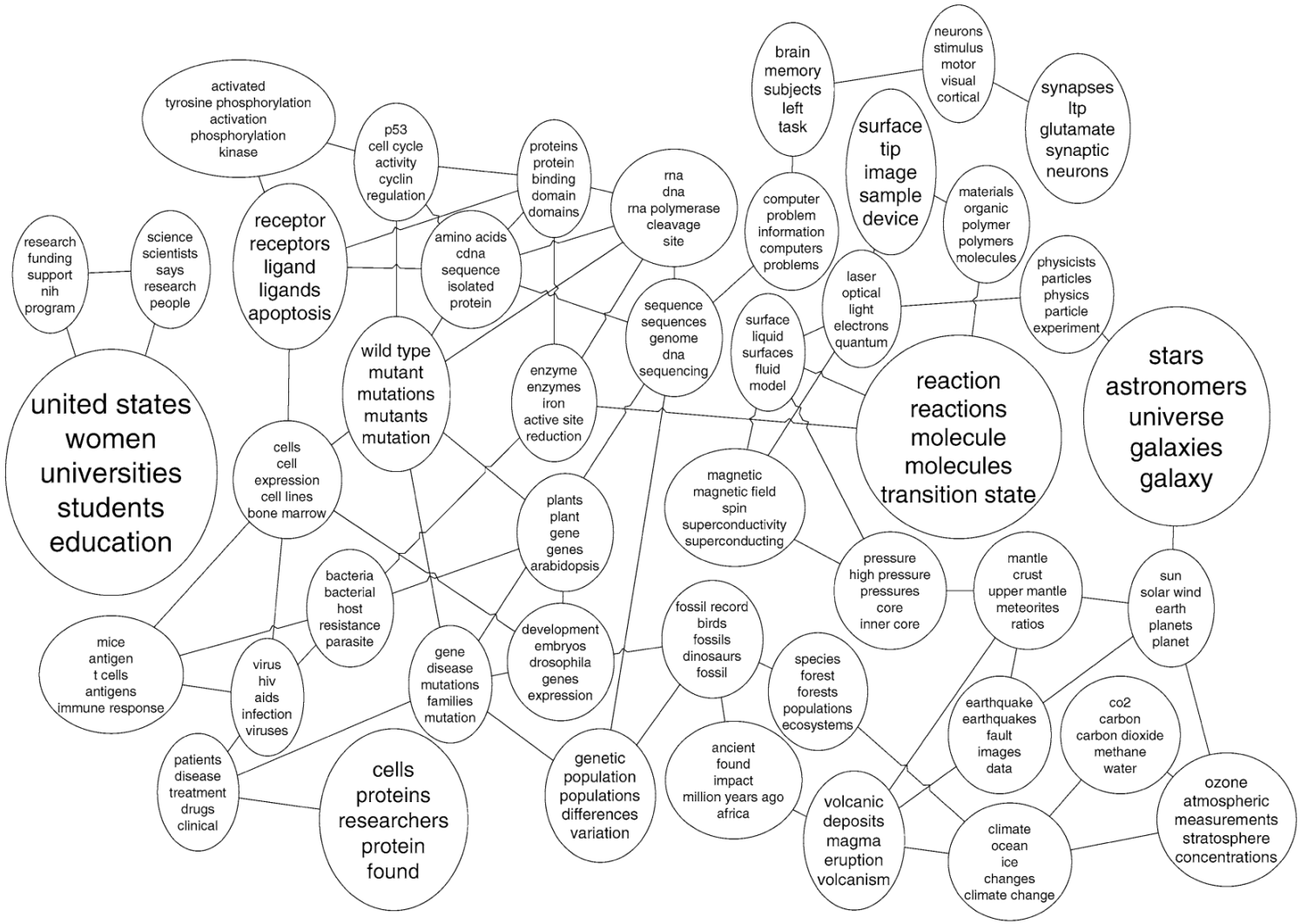


FIG. 2. *A portion of the topic graph learned from* 16,351 *OCR articles from Science* (1990–1999). *Each topic node is labeled with its five most probable phrases and has font proportional to its popularity in the corpus.* (*Phrases are found by permutation test.*) *The full model can be found in* http://www.cs.cmu.edu/~lemur/science/ *and on STATLIB.*

Topics being discussed in the Senate

Skip

TABLE 3 Topic Keywords for 42-Topic Model

Topic (Short Label)	Keys
1. Judicial Nominations	*nomine, confirm, nomin, circuit, hear, court, judg, judici, case, vacanc*
2. Constitutional	*case, court, attornei, supreme, justic, nomin, judg, m, decis, constitut*
3. Campaign Finance	*campaign, candid, elect, monei, contribut, polit, soft, ad, parti, limit*
4. Abortion	*procedur, abort, babi, thi, life, doctor, human, ban, decis, or*
5. Crime 1 [Violent]	*enforc, act, crime, gun, law, victim, violenc, abus, prevent, juvenil*
6. Child Protection	*gun, tobacco, smoke, kid, show, firearm, crime, kill, law, school*
7. Health 1 [Medical]	*diseas, cancer, research, health, prevent, patient, treatment, devic, food*
8. Social Welfare	*care, health, act, home, hospit, support, children, educ, student, nurs*
9. Education	*school, teacher, educ, student, children, test, local, learn, district, class*
10. Military 1 [Manpower]	*veteran, va, forc, militari, care, reserv, serv, men, guard, member*
11. Military 2 [Infrastructure]	*appropri, defens, forc, report, request, confer, guard, depart, fund, project*
12. Intelligence	*intellig, homeland, commiss, depart, agenc, director, secur, base, defens*
13. Crime 2 [Federal]	*act, inform, enforc, record, law, court, section, crimin, internet, investig*
14. Environment 1 [Public Lands]	*land, water, park, act, river, natur, wildlif, area, conserv, forest*
15. Commercial Infrastructure	*small, busi, act, highwai, transport, internet, loan, credit, local, capit*
16. Banking / Finance	*bankruptci, bank, credit, case, ir, compani, file, card, financi, lawyer*
17. Labor 1 [Workers]	*worker, social, retir, benefit, plan, act, employ, pension, small, employe*
18. Debt / Social Security	*social, year, cut, budget, debt, spend, balanc, deficit, over, trust*
19. Labor 2 [Employment]	*job, worker, pai, wage, economi, hour, compani, minimum, overtim*
20. Taxes	*tax, cut, incom, pai, estat, over, relief, marriag, than, penalti*
21. Energy	*energi, fuel, ga, oil, price, produce, electr, renew, natur, suppli*
22. Environment 2 [Regulation]	*wast, land, water, site, forest, nuclear, fire, mine, environment, road*
23. Agriculture	*farmer, price, produc, farm, crop, agricultur, disast, compact, food, market*
24. Trade	*trade, agreement, china, negoti, import, countri, worker, unit, world, free*

Topics being discussed in the Senate

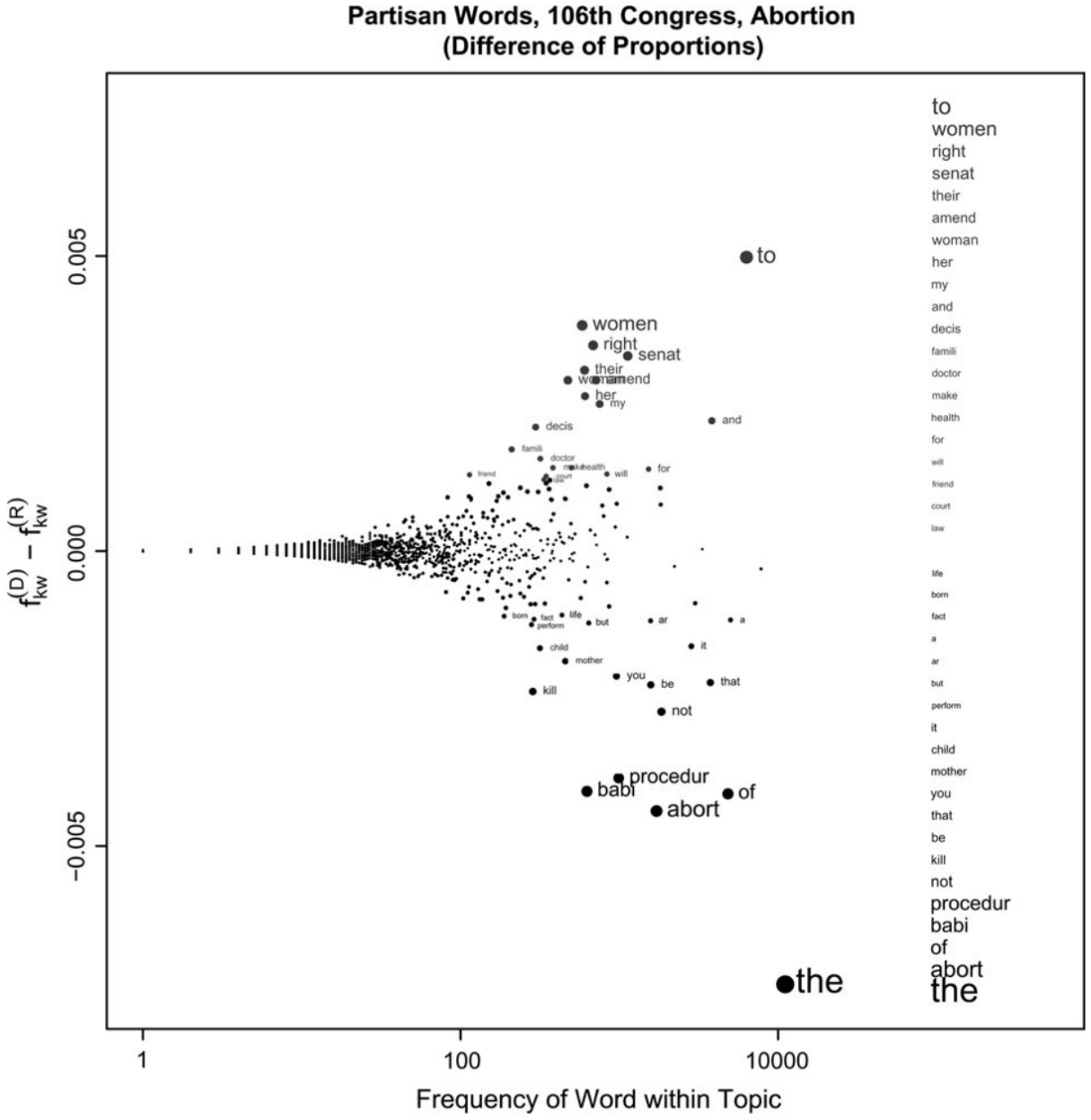
Skip

25. Procedural 3	*mr, consent, unanim, order, move, senat, ask, amend, presid, quorum*
26. Procedural 4	*leader, major, am, senat, move, issu, hope, week, done, to*
27. Health 2 [Seniors]	*senior, drug, prescript, medicar, coverag, benefit, plan, price, beneficiari*
28. Health 3 [Economics]	*patient, care, doctor, health, insur, medic, plan, coverag, decis, right*
29. Defense [Use of Force]	*iraq, forc, resolut, unit, saddam, troop, war, world, threat, hussein*
30. International [Diplomacy]	*unit, human, peac, nato, china, forc, intern, democraci, resolut, europ*
31. International [Arms]	*test, treati, weapon, russia, nuclear, defens, unit, missil, chemic*
32. Symbolic [Living]	*serv, hi, career, dedic, john, posit, honor, nomin, dure, miss*
33. Symbolic [Constituent]	*recogn, dedic, honor, serv, insert, contribut, celebr, congratul, career*
34. Symbolic [Military]	*honor, men, sacrific, memori, dedic, freedom, di, kill, serve, soldier*
35. Symbolic [Nonmilitary]	*great, hi, paul, john, alwai, reagan, him, serv, love*
36. Symbolic [Sports]	*team, game, plai, player, win, fan, basebal, congratul, record, victori*
37. J. Helms re: Debt	*hundr, at, four, three, ago, of, year, five, two, the*
38. G. Smith re: Hate Crime	*of, and, in, chang, by, to, a, act, with, the, hate*
39. Procedural 1	*order, without, the, from, object, recogn, so, second, call, clerk*
40. Procedural 5	*consent, unanim, the, of, mr, to, order, further, and, consider*
41. Procedural 6	*mr, consent, unanim, of, to, at, order, the, consider, follow*
42. Procedural 2	*of, mr, consent, unanim, and, at, meet, on, the, am*

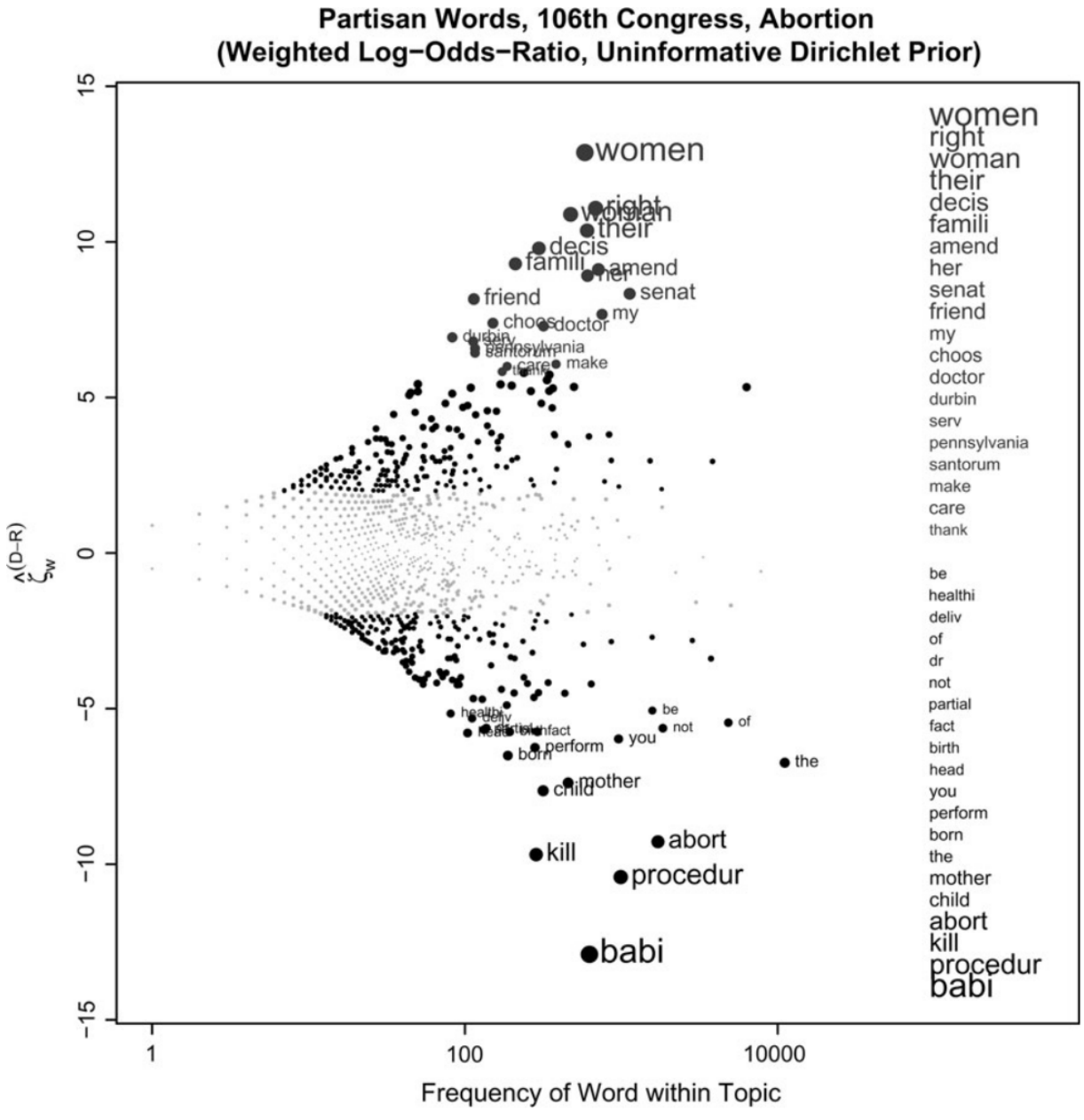
Notes: For each topic, the top 10 (or so) key stems that best distinguish the topic from all others. Keywords have been sorted here by $\text{rank}(\beta_{kw}) + \text{rank}(r_{kw})$, as defined in the text. Lists of the top 40 keywords for each topic and related information are provided in the web appendix. Note the order of the topics is the same as in Table 2 but the topic names have been shortened.

Partisanship in political debates

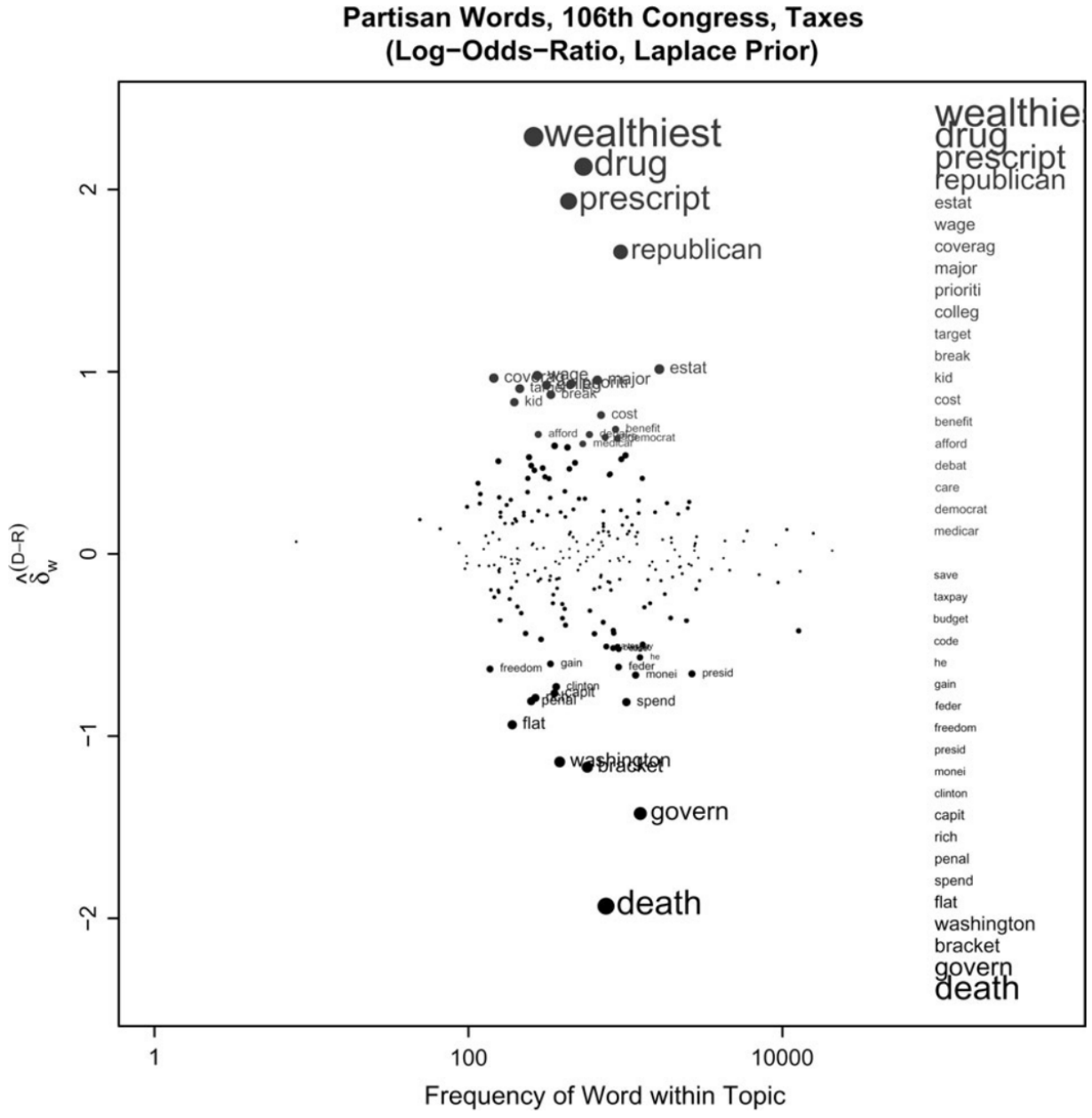
Skip



Partisanship in political debates



Partisanship in political debates



Partisanship in political debates

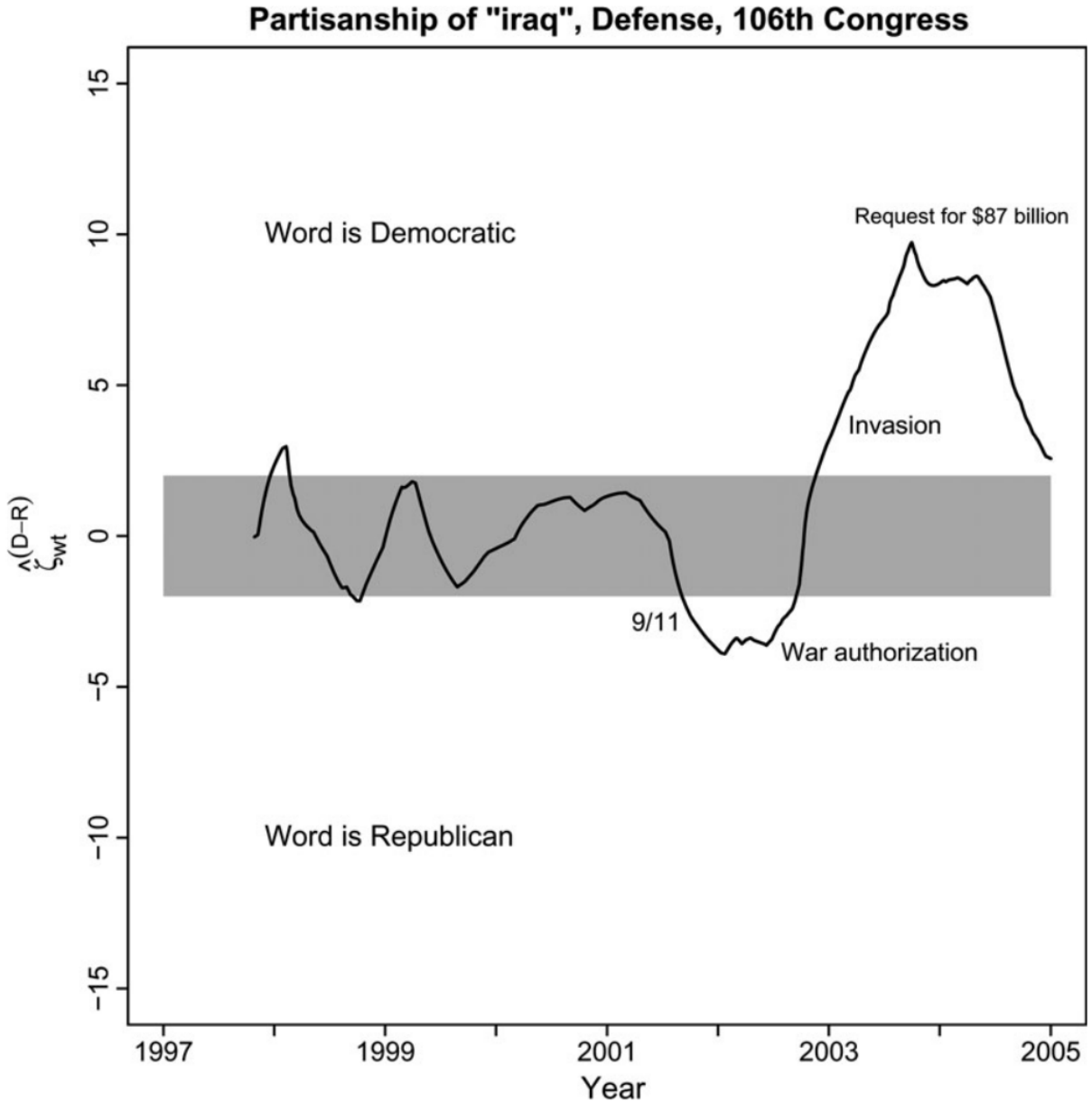
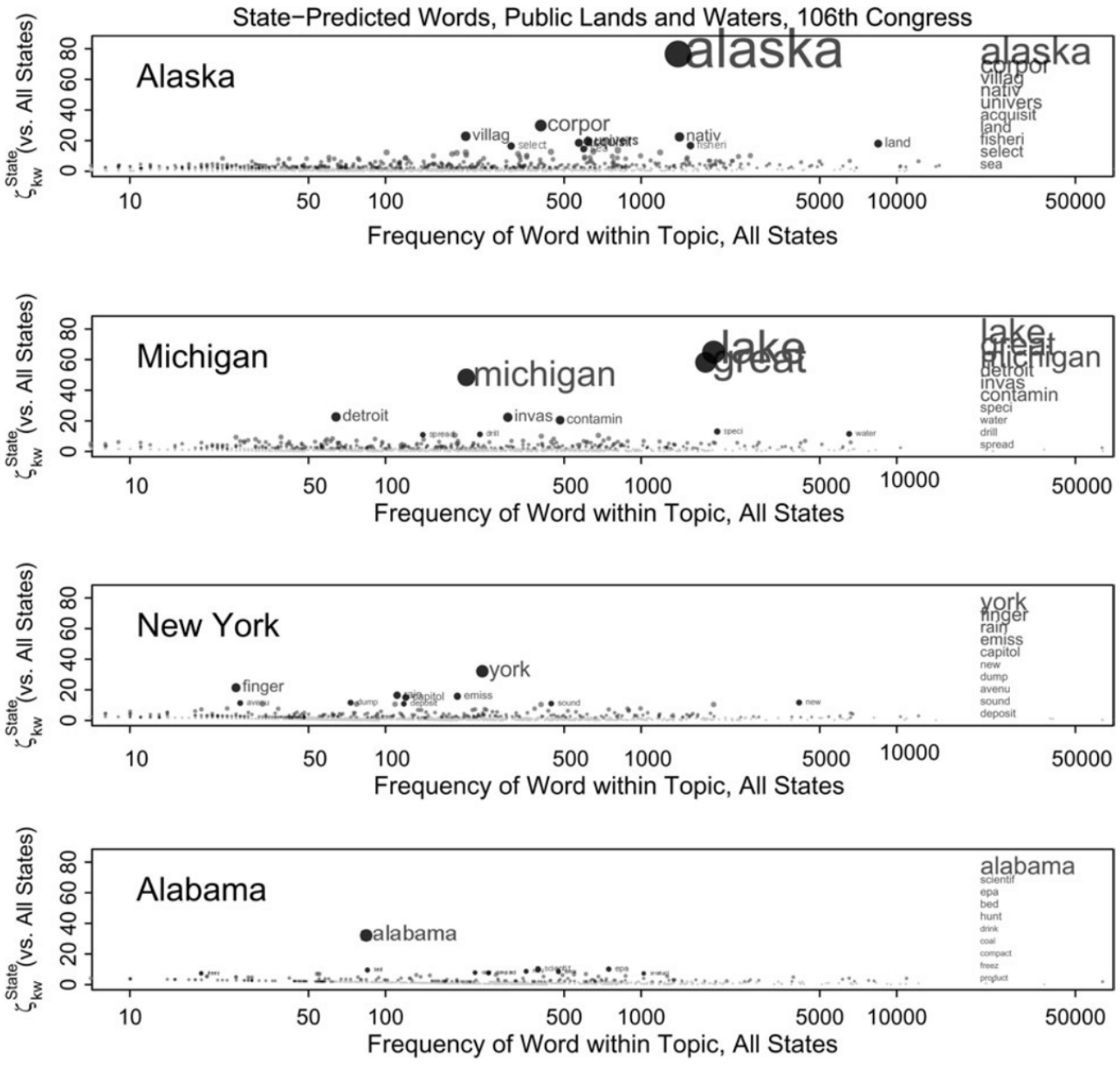


Fig. 9 Dynamic partisanship of "iraq" in the context of defense.

Partisanship in political debates



Practical considerations

Practical considerations

- Dictionary methods are based on prior information about the function mapping features c_i to outcomes v_i.
- Use them when such information is strong and corresponding information in the data weak.
- Example: Outcomes never observed, and the mapping of interest is not picked up by the factor structure of unsupervised methods.
- Or: Training data exists, but is too small and noisy such that a prior-driven specification of $f(\cdot)$ is deemed more reliable.

Practical considerations

- Text regression is good for predicting a single attribute, especially when a large amount of training data is available.
- Supervised generative techniques like naive Bayes or MNIR may improve prediction when p is large relative to n, but gains diminish rapidly with sample size.
- In text regression it is typically unnecessary to learn flexible functional forms (and unwise unless $n \gg p$).
- This is why linear index regression methods work well and are popular.
- Most prediction tasks with text input in social science are efficiently addressed via penalized linear regression.

Practical considerations

- If there are multiple attributes of interest and interdependence between attributes and their effect on language should be resolved, resort to generative models.
- Use topic modeling for corpora of many unlabeled documents.
- Caution with the interpretation of unsupervised methods (multimodal distributions of parameter estimates). The best way to build interpretability for topic models is to add some supervision.

Practical considerations

- Methods can always be combined.
- Use generative model to transform inputs, then apply supervised methods.
- Not limited to linear index regression, explore other methods: tree-based estimators, SVM, boosting, ...
- Value of generative models in *dimensionality reduction*.
- Objective: Interpretability or predictive power?

Model validation

- Prediction: Cross-validation, reserving a test set.
- Tuning for multi-step modeling can be expensive. Nested CV for dimension reduction step and modeling step.
- Descriptive or causal analysis: Validate the accuracy with which the fitted model is capturing the quantity of interest.
- *Manual audits* are effective. Check some subset of fitted values against the coding a human would produce by hand.
- Especially relevant for dictionary methods: Validity hinges on particular keywords. If you have sufficient prior information to justify this, you should have sufficient information to evaluate whether the resulting classification is accurate.
- Audits can be formalized with multiple people and quantified consistency checks.
- Inspection of estimated parameters may (more likely may not) be informative.

Next lecture: Distributional language models.

References

- Blei, D. M. (2012). Probabilistic Topic Models. *Communications of the ACM* 55(4), 77–84. DOI: 10.1145/2133806.2133826.
- Blei, D. M. and J. D. Lafferty (2007). A Correlated Topic Model of Science. *Annals of Applied Statistics* 1(1), 17–35. DOI: 10/bzkc9z.
- Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3(Jan), 993–1022.
- Butler, D. (2013). When Google Got Flu Wrong. *Nature News* 494(7436), 155. DOI: 10/kg9.
- Choi, H. and H. R. Varian (2012). Predicting the Present with Google Trends. *Economic Record* 88(s1), 2–9. DOI: 10/gf658w.
- Cowles, A. (1933). Can Stock Market Forecasters Forecast? *Econometrica* 1(3), 309–324. DOI: 10.2307/1907042.
- Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39(1), 1–22. DOI: 10.1111/j.2517-6161.1977.tb01600.x.

References

Forgy, E. W. (1965). Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classifications. *biometrics* 21, 768–769.

Gentzkow, M. and J. M. Shapiro (2010). What Drives Media Slant? Evidence From U.S. Daily Newspapers. *Econometrica* 78(1), 35–71. DOI: 10/br3763.

Ginsberg, J., M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant (2009). Detecting Influenza Epidemics Using Search Engine Query Data. *Nature* 457(7232), 1012–1014. DOI: 10/fmmjzc.

Grimmer, J. and B. M. Stewart (2013). Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis* 21(3), 267–297. DOI: 10/f458q9.

Groseclose, T. and J. Milyo (2005). A Measure of Media Bias. *The Quarterly Journal of Economics* 120(4), 1191–1237. DOI: 10/cz3n76.

Klebanov, B. B., D. Diermeier, and E. Beigman (2008). Lexical Cohesion Analysis of Political Speech. *Political Analysis* 16(4), 447–463. DOI: 10/bhrdb8.

Lauderdale, B. E. and A. Herzog (2016). Measuring Political Positions from Legislative Speech. *Political Analysis* 24(3), 374–394. DOI: 10/f873xn.

References


Laver, M. and J. Garry (2000). Estimating Policy Positions from Political Texts. *American Journal of Political Science* 44(3), 619–634. DOI: 10/fp84jh.


Lazer, D., R. Kennedy, G. King, and A. Vespignani (2014). The Parable of Google Flu: Traps in Big Data Analysis. *Science* 343(6176), 1203–1205. DOI: 10/rwx.


MacQueen, J. (1967). "Some Methods for Classification and Analysis of Multivariate Observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. The Regents of the University of California.


Monroe, B. L., M. P. Colaresi, and K. M. Quinn (2008). Fightin' Words: Lexical Feature Selection and Evaluation for Identifying the Content of Political Conflict. *Political Analysis* 16(4), 372–403. DOI: 10/cb486t.


Mosteller, F. and D. L. Wallace (1963). Inference in an Authorship Problem. *Journal of the American Statistical Association* 58(302), 275–309. DOI: 10/c57r6p.

References


Mosteller, F. and D. L. Wallace (1964). *Inference and Disputed Authorship: The Federalist*. David Hume series ed. The David Hume Series. Stanford, Calif: Center for the Study of Language and Information.


Ng, A. Y. and M. I. Jordan (2002). On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes. In: *Advances in Neural Information Processing Systems 14*. Ed. by T. G. Dietterich, S. Becker, and Z. Ghahramani. MIT Press, 841–848.


Quinn, K. M., B. L. Monroe, M. Colaresi, M. H. Crespin, and D. R. Radev (2010). How to Analyze Political Attention with Minimal Assumptions and Costs. *American Journal of Political Science* 54(1), 209–228. DOI: 10/cvfs44.


Roberts, M. E., B. M. Stewart, D. Tingley, C. Lucas, J. Leder-Luis, S. K. Gadarian, B. Albertson, and D. G. Rand (2014). Structural Topic Models for Open-Ended Survey Responses. *American Journal of Political Science* 58(4), 1064–1082. DOI: 10/f6mv8k.

References

Saiz, A. and U. Simonsohn (2013). Proxying for Unobservable Variables with Internet Document-Frequency. *Journal of the European Economic Association* 11(1), 137–165. DOI: 10.1111/j.1542-4774.2012.01110.x.

Scott, S. L. and H. R. Varian (2014). Predicting the Present with Bayesian Structural Time Series. *International Journal of Mathematical Modelling and Numerical Optimisation* 5(1-2), 4–23. DOI: 10/gg9fmw.

Scott, S. L. and H. R. Varian (2015). *Bayesian Variable Selection for Nowcasting Economic Time Series*. University of Chicago Press. Chap. Economic Analysis of the Digital Economy.

Slapin, J. B. and S.-O. Proksch (2008). A Scaling Model for Estimating Time-Series Party Positions from Texts. *American Journal of Political Science* 52(3), 705–722. DOI: 10/brh9q7.

Steinhaus, H. (1956). Sur La Division Des Corps Materiels En Parties. Bull. Acad. Polon. Sci., C1. III Vol IV: 801-804.

References

Stephens-Davidowitz, S. (2014). The Cost of Racial Animus on a Black Candidate: Evidence Using Google Search Data. *Journal of Public Economics* 118, 26–40. DOI: 10/f6j3vk.

Stock, J. H. and F. Trebbi (2003). Retrospectives Who Invented Instrumental Variable Regression? *The Journal of Economic Perspectives* 17(3), 177–194. DOI: 10/d796t2.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 58(1), 267–288.

Vannoni, M., E. Ash, and M. Morelli (2020). Measuring Discretion and Delegation in Legislative Texts: Methods and Application to US States. *Political Analysis*, 1–15. DOI: 10/gg9fmv.

Varian, H. R. (2014). Big Data: New Tricks for Econometrics. *Journal of Economic Perspectives* 28(2), 3–28. DOI: 10/gc3fvv.

Wright, P. G. (1928). *The Tariff on Animal and Vegetable Oils, by Philip G. Wright, with the Aid of the Council and Staff of the Institute of Economics. [Preface by Harold G. Moulton.]*. Macmillan Company.