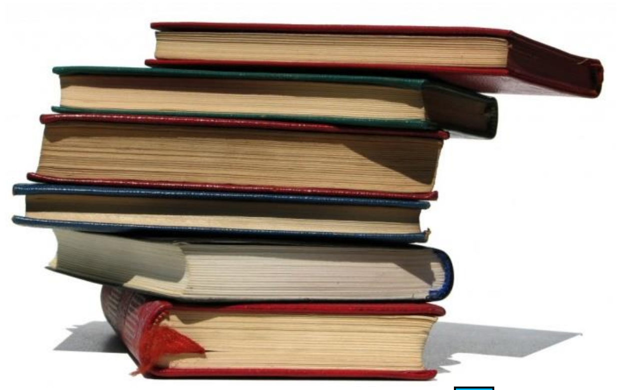
Big Data Analytics

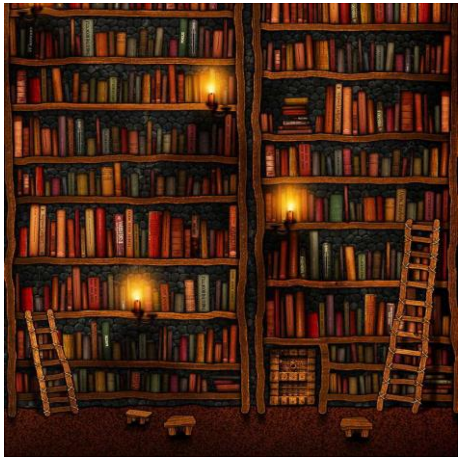
Lecture 2

Unsupervised scaling algorithms: Wordfish

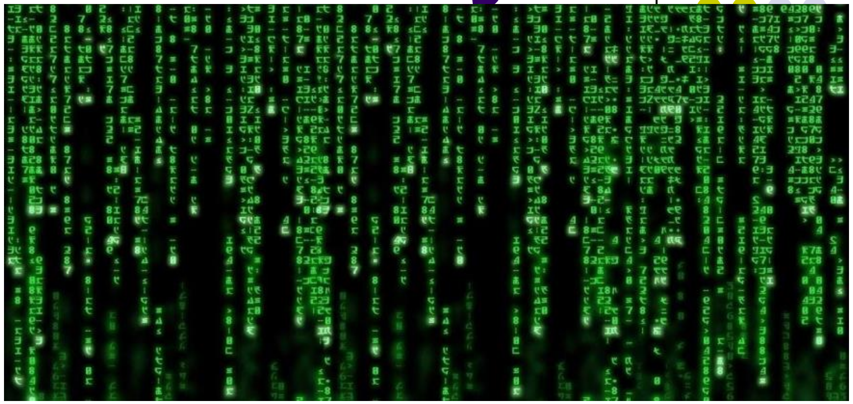
But before that…a summary



You pass to R the texts you want to analyze via `readtext`



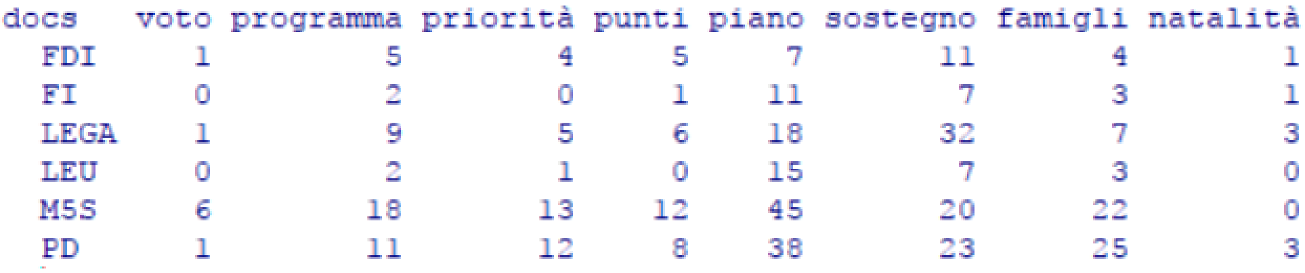
You tell to R that those bunch of texts belong to the *same collection of texts* you want to analyze via `corpus`



You extract from the corpus the relative document term (or feature) matrix via `dfm`. By doing that you apply the bag of words approach to that corpus of texts

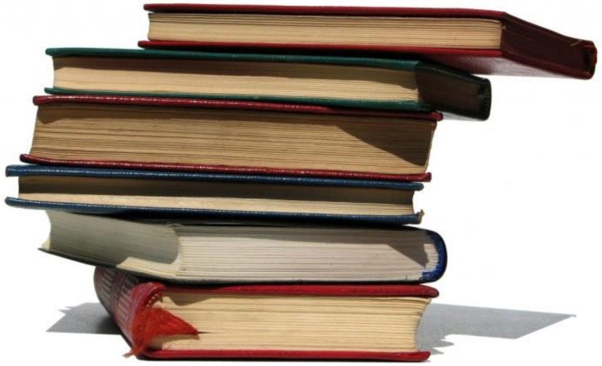
But before that…a summary

All the statistical models that we will see, **work on this**

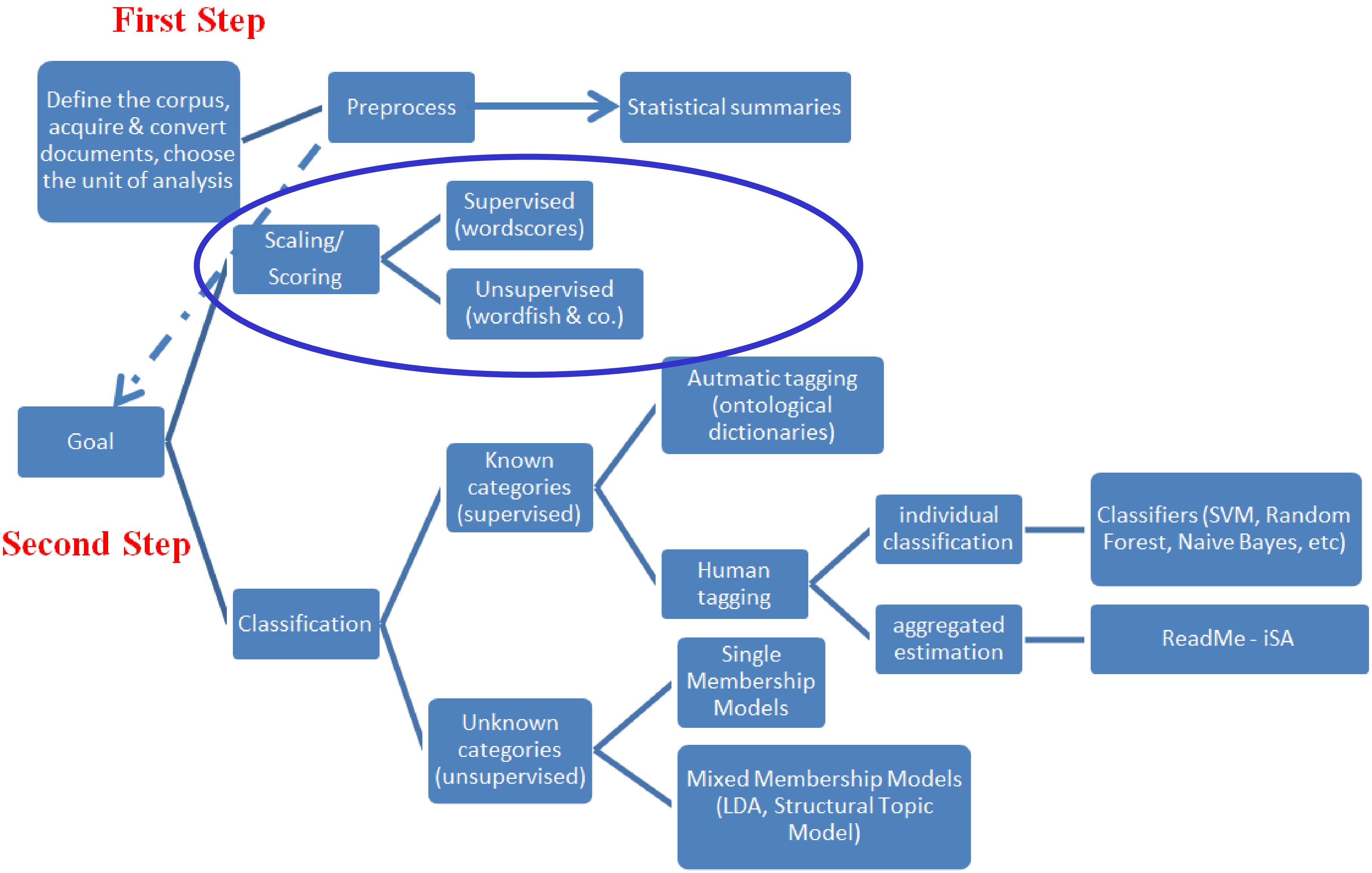


```
docs    voto programma priorità punti piano sostegno famigli natalità
  FDI      1         5        4     5     7       11       4        1
  FI       0         2        0     1    11        7       3        1
  LEGA     1         9        5     6    18       32       7        3
  LEU      0         2        1     0    15        7       3        0
  M5S      6        18       13    12    45       20      22        0
  PD       1        11       12     8    38       23      25        3
```

NOT on this



Our Course Map



First Step

Define the corpus, acquire & convert documents, choose the unit of analysis

Preprocess

Statistical summaries

Scaling/ Scoring

Supervised (wordscores)

Unsupervised (wordfish & co.)

Goal

Second Step

Classification

Known categories (supervised)

Autmatic tagging (ontological dictionaries)

Human tagging

individual classification

Classifiers (SVM, Random Forest, Naive Bayes, etc)

aggregated estimation

ReadMe - iSA

Unknown categories (unsupervised)

Single Membership Models

Mixed Membership Models (LDA, Structural Topic Model)

References

- Proksch, Sven-Oliber, and Slapin, Jonathan B. 2008. "A Scaling Model for Estimating Time-Series Party Positions from Texts". *American Journal of Political Science*, 52(3): 705-722.
- Proksch, Sven-Oliber, and Slapin, Jonathan B. 2009. "How to Avoid Pitfalls in Statistical Analysis of Political Texts: The Case of Germany". *German Politics*, 18(3): 323-344
- Curini, Luigi, Airo Hino, and Atsushi Osaki. 2018. "Intensity of government–opposition divide as measured through legislative speeches and what we can learn from it. Analyses of Japanese parliamentary debates, 1953–2013". *Government and Opposition*, DOI: 10.1017/gov.2018.15

Latent models

Textual data might focus on **manifest characteristics** whose significance lies primarily in **how they were communicated** in the text

To take an example, if we were interested in whether a political speaker used **racist language**, this language **would be manifest directly in the text itself** in the form of racist terms or references, and what would matter is **whether they were use**d, not so much **what they might represent**

Latent models

However, sometimes the target of concern is not so much **what the text contains**, but what **its contents reveal as data about the latent characteristics** for which the text provides ***observable implications***

Is this important? YES!

Latent models

In the study of politics, important theories about political and social actors concern qualities that are **unobservable through direct means**

Ideology, in particular, is fundamental to the study of political competition and political preferences, but we have **no direct measurement instrument** for recording an individual or party’s relative preference for (for example) liberal policies versus conservative ones

That is, ideology is not something that the researcher can **directly observe**…rather it must be indirectly estimated based upon **observable actions** taken by actors

Latent models

Observable actions…such as?

Roll-calls, for example! Still, voting in a legislature is subject to **party discipline** and may be **highly strategic**….and so?

Let’s rely on something else then…like…what?!?

Texts, of course!

A big advantage: estimating positions using existing data is often difficult and sometimes impossible…but nearly all political actors speak (or write)!

Scaling methods

The goal of methods **for scaling positions** is to use **some observed set of outcomes** to draw inferences about an actor’s (in the widest sense of the word) unobservable position on a **latent dimension** *relative* to other actors

Position is here to be understood as a **preference on that dimension**. To get at such a position, the **observed outcomes** must reveal some kind of preference on the part of the actor

Scaling methods

When scaling the political positions of a corpus of texts, we can view the **choice of words as the observed outcome**

Whenever **certain statements are associated with particular preferences**, we can use them to discriminate between positions as expressed in different documents along a certain continuous space (uni- or multi-dimensional)

In other words, the use of a particular (set of) word(s) provides us with **revealed preferences** that could be related to ideology, or to some other policy (or non-policy) space

Scaling methods

Scaling methods can be differentiated between **Supervised & Unsupervised Methods**

Supervised models use human input, typically in the form of a set of reference texts that have been already validated (i.e., **already classified** as left, right, extreme right texts for example)

These estimates can then be used to predict (i.e., scale) the positions of texts the model has not encountered previously (i.e., virgin texts)

The reference texts also serves to **define the content of the space** that the researcher seeks to estimate (if you use a set of reference-texts validated over a left-right economic scale, you will scale the virgin texts along such scale. More on this later on)

Scaling methods

Unsupervised Methods simultaneously learn about the latent space and estimate document positions in it, without any input from the researcher, i.e., they "*discover*" words that distinguish locations on some dimensional spectrum (**not defined a-priori** as it happens in the case of the supervised scaling methods)

How is possible? Give me a moment…

…first we need to discussion about two assumptions!

Scaling methods

Which Assumptions are needed to Scale a Text?

First, if the cost to articulating a position is low, authors' might engage in *cheap talk*

Conversely, if costs are high, they might choose not to articulate the position for *strategic reasons*

All the scaling techniques we focus on, assume on the contrary that authors do **not censor their statements for political reasons**

This assumption, in some given circumstances, could however cause significant measurement error

Scaling methods

Scaling methods

Second, the documents **should be informative about the differences** we seek to estimate

Particularly in contexts where there are **strong common norms about how to phrase a documen**t (as with highly technical legislative or legal documents) or the texts do not communicate any preference at all, it can be difficult to scale documents

If authors of different preferences use similar choices of words, we cannot in fact use the texts to discriminate between their positions

Wordfish

Unsupervised methods for scaling texts produce estimates using **only the information available** in the textual data itself

How to do that?

Let’s introduce **Wordfish**!

Wordfish

Wordfish assumes that the **language** used by political actors expresses political preferences, that is…

… political preferences manifest themselves in the **word choice** of politicians when writing party documents or saying something for example

Wordfish

More specifically, Wordfish assumes that **relative word usage** within documents conveys information about their positions in some policy space

To give an example, the technique assumes that if one party uses the word ‘**freedom**’ more frequently than the word ‘**equality**’ in a document on economic policy while another party uses ‘equality’ more often than ‘freedom’ in a similar document, these **two words** – ‘equality’ and ‘freedom’ – **provide information** about party preferences with regard to an underlying policy dimension, and **discriminate** between the parties

Scaling methods and theory

The issue on “relative emphasis” makes (political) sense!

The assumption behind models like Wordfish in fact is to some degree based on the Saliency Theory as applied in the Comparative Manifesto Project (Budge et al., 2001)

According to such theory “policy differences between parties consist of contrasting emphases on different policy areas (thus, one party often mentions taxes, another welfare, etc.)”

That is, it is the “relative emphasis” of one word (or category) over another that **signals position**

Wordfish

Note one important aspect: the interpretation of the **estimated dimension** in Wordfish **is completely left to the researcher**

In the previous example, Wordfish **does not tell the researcher** whether ‘equality’ is a ‘left-wing word’ while ‘freedom’ is a ‘right-wing word’

The algorithm will simply use the relative frequencies of these words as data to locate the documents on a latent continuous scale, and it is up to the researcher to make an assessment about what constitutes ‘left’ and ‘right’ based upon her **knowledge of politics** (*a-posteriori* method!)

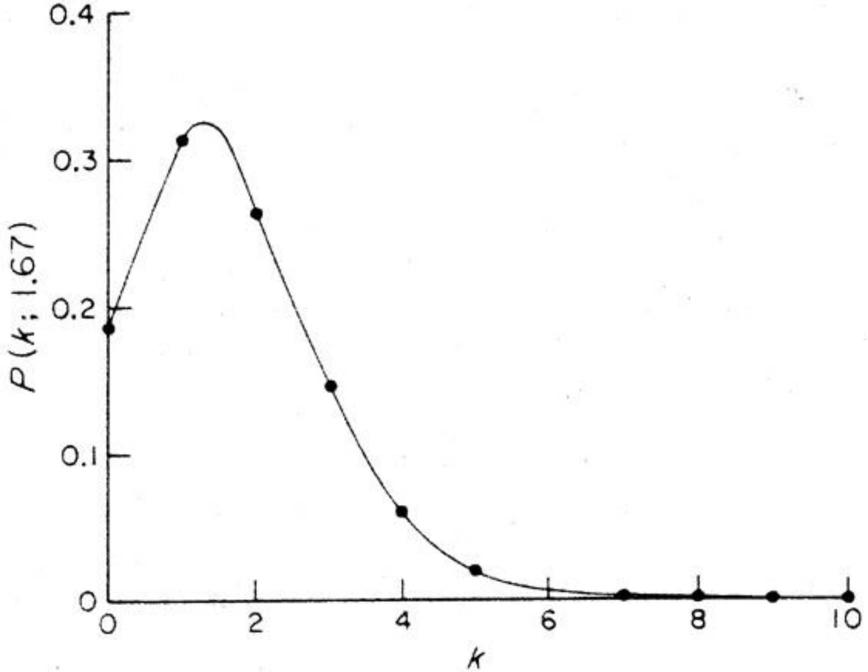
Wordfish Estimation Process

The *discover* of words that distinguish locations on a political spectrum is made possible by adopting some statistical assumptions on the **distribution of words** employed in texts

Wordfish Estimation Process

But which is the **statistical distribution** which most **accurately approximate word usage**?

Wordfish assumes that word frequencies (the number of times an actor *i* mentions word *j)* are generated by/drawn from a **Poisson process**, a distribution that is **heavily skewed**, as is the case of **word usage**



More formally

Formally, the functional form of the model is as follows: $y_{ijt} \approx POISSON(\lambda_{ijt})$ where y_{ijt} is the **count** of word *j* in document *i*'s (i.e., party manifesto; speech; etc.) at time *t*

The lambda parameter has the following systematic component:

$$\lambda_{ijt} = exp(\alpha_{it} + \psi_j + \beta_j * \theta_{it})$$

Wordfish Estimation Process

The **systematic component** of this process contains 4 parameters: 1) *word fixed effects* Ψ (psi); 2) *document fixed effects at time t* α; 3) *document positions* θ at time *t* (theta); 4) *word weights* β

Word fixed effects are included to capture the fact that some words need to be used **much more often** in a language

Such words may serve a grammatical purpose but they have no substantive or ideological meaning, such as conjunctions or definite and indefinite articles

Wordfish Estimation Process

The **document fixed effect** parameters control for the possibility that some documents in the analysis may be **significantly longer** than others

When using manifestos to estimate party positions, for example, this can happen when some parties in some years write much longer manifestos

Wordfish Estimation Process

The **document positions parameters** tells us the positions of each document relative to the other documents in the recovered latent space

Finally, the **word discrimination parameters** allow the researcher to analyze **which words differentiate documents (party) positions**

This allows the researcher to estimate party positions and uncover the variations in political language that are responsible for placing parties on this dimension

Wordfish Estimation Process

Let’s see one example

In Curini et al. (2018), we have selected all the speeches in which Japanese Prime Ministers make a general policy speech (*shoshin hyoumei enzetsu*) in the following situations:

i) after being nominated in the Special session

ii) after having succeeded a predecessor during a parliamentary session

iii) and in the beginning of the Extraordinary session

Wordfish Estimation Process

Overall 439 speeches over 82 sessions, and almost 20,000 words/kanji

URL to get access to Japanese legislative speeches:

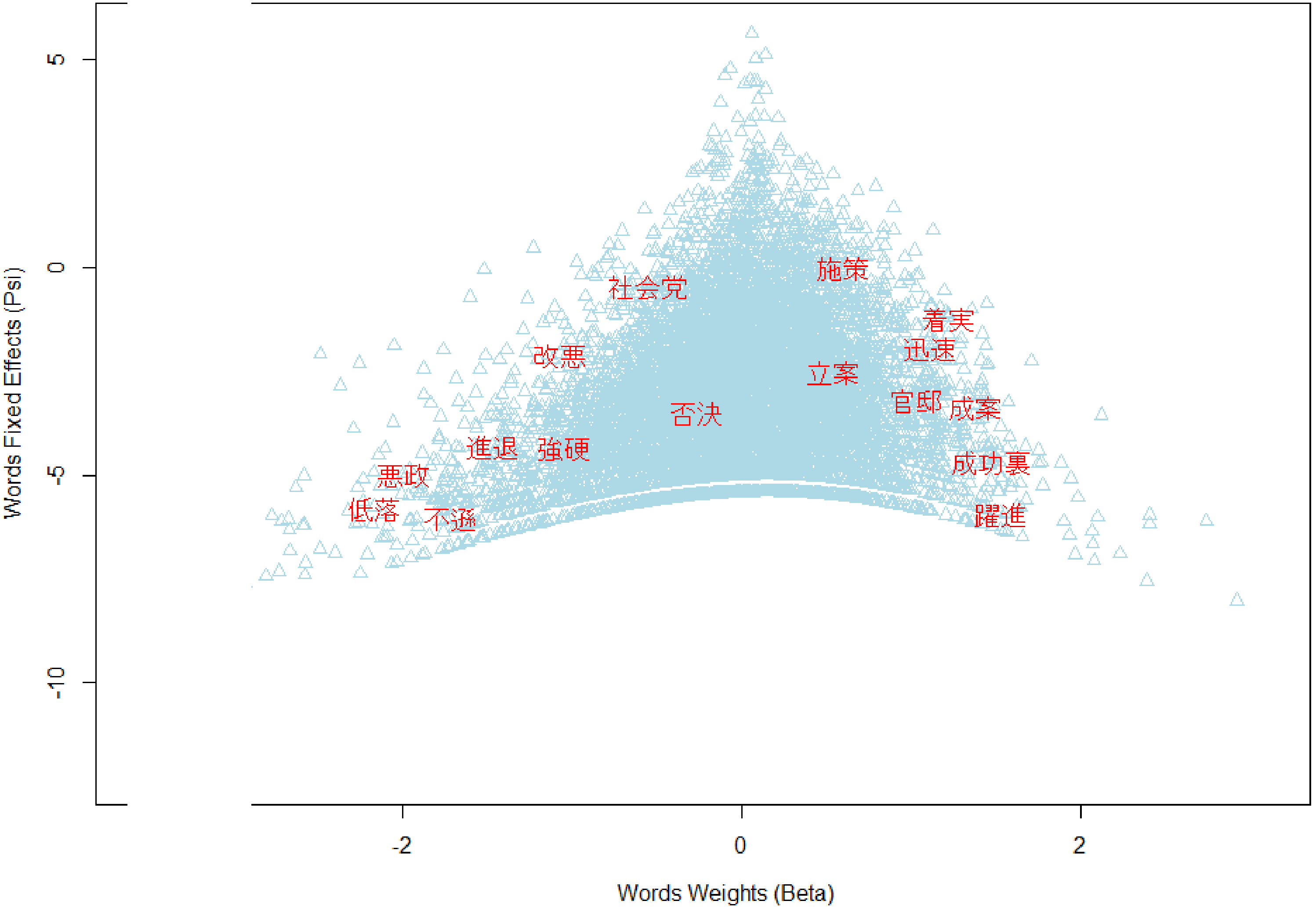
http://kokkai.ndl.go.jp/

Of course, we **tokenized** all the texts!!!

Our time range: 1953/2013 (pretty long period…more on this below…)

The discriminating words

Diagnostics of word's estimates: 1953-2013



The discriminating words

Positive betas: *breakthrough, successfully, bills passed, steady, prompt, policy measure, policy making*

Negative betas: *decline, misgovernment, arrogance, decision to leave from a position, deterioration, by force, rejecting bills*

What we have to do is therefore **linking** the discriminating words parameters β with the documents' position θ parameters

The discriminating words

In the example just saw, *bills passed* has a high absolute positive value for its discrimination value. Therefore, party's documents using that words with high frequency will receive a positive score along the latent dimension (cabinet parties?)

The word *rejecting bills* would also have a high absolute value **but with the opposite (negative) sign**. Therefore, party's documents using that words with high frequency will receive a negative score along the latent dimension (opposition parties?)

Therefore the latent dimension is a *opposition-cabinet one*?

More formally

WORDFISH uses an **expectation maximization (EM) algorithm** to retrieve maximum likelihood estimates for all parameters

The implementation of this algorithm entails an **iterative process**:

first *document parameters* are held fixed at a certain value while *word parameters* are estimated, **then** *word parameters* are held fixed at their new values while the *document parameters* are estimated

This process is **repeated until the parameter estimates** reach an acceptable level of convergence

Some challenges

1. Document processing
2. Interpretation
3. Dynamic pattern

Document Processing

Document processing is essential and possibly the most tricky task in the estimation process in Wordfish (and not only for this method…)

Researchers should define the set of texts to be analyzed

A pre-requirement: the model specification used by Wordfish works best as **more data is available**, meaning as **more documents (and more words)** are used in the analysis

So, using Wordfish to scale for example tweets (i.e., very short texts) is not a great idea…

Document Processing

Having said that, the selection of texts will crucially depend on what kind of **dimension** should be analyzed

Wordfish estimates a **single dimension**, and the information contained in this **dimension depends upon the texts** that the researcher chooses to analyze

Therefore, the **selection of texts should depend** on the particular dimension the researcher wishes to examine

Document Processing

For instance, if a researcher is interested in comparing **foreign policy statements** of parties in country X, then only such texts should be included in the analysis

On the other hand, if the research question is to determine a **general ideological position** using all aspects of policy (e.g. left-right), then the analysis should potentially be conducted using all parts of an election manifesto, for example, assuming that such documents are encyclopedic statements of policy positions

Document Processing

The estimated single dimension **will thus be a function** of the selection of the text corpus

This also implies that when the generative model specifies a unidimensional policy space, when it really is *multidimensional*, we risk miss-specifying the dimension we extract! **Why?**

Document Processing

Wordfish will recognize differences in word use between two texts as indicative of their different political positions

These differences could be however also due to the topics addressed by the authors , i.e., situations where texts do not address **similar topics at all**

In these situations texts cannot be reasonably scaled together, and if they are, it will often result in the main latent dimension being grossly miss-specified

Document Processing

For example, if you have a set of texts discussing about K-pop and a set of texts discussing about Japanese politics, and you scale them together…

…you will obtain a latent scale that will differentiate between K-pop texts on one extreme of the latent dimension and texts discussing about Japanese politics on the other extreme. What’s the utility of that?

Document Processing

WORDFISH does not estimate **multiple dimensions**, but it does allow the estimation of **different dimensions** if you use different text sources

For instance, if your interest is in estimating positions of **presidential candidates on foreign policy and economic policy**, then you could estimate separate positions using foreign policy speeches only on the one hand and economic policy speeches on the other hand and from this creating a **2-dimensional space**

Document Processing

How can we be sure that a single dimension is a good approximation of the underlying latent competition in the texts that we are analyzing?

We cannot! Theory and face validity of the results can help

It also can help (in some given circumstances) comparing the results you get via Wordfish with the results you get via Correspondence Analysis

If you are interest about, just drop me an email

Interpretation

Position estimates derived using Wordfish are based **only on the information in the texts**

This lack of an *ex-ante* defined dimensionality is a **double-edged sword**: while Wordfish scales texts independently of prior information, it renders **uncertain** the exact nature of the dimension being estimated (as it happens in all unsupervised approaches!)

One important drawback of unsupervised algorithms is thus that the nature of the dimensions produced requires **intensive validation** before they can be applied across different sets of texts and contexts

Interpretation

Quite often papers rely on the strong assumption of **ideological dominance in speech** (i.e., that actors' ideological leanings determine what is discussed in texts)…sometimes this makes sense, other times no!

This is **not a shortcoming** of Wordfish!

This simply suggests that one **should not blindly assume** that Wordfish output measures an ideological location of political actors without careful validation

In the previous example about Japan, we actually capture an opposition-cabinet latent dimension!

Dynamic Estimation

Using text to estimate party positions **over time** creates an additional challenge. On the one hand, we would like to use as much information in the texts as possible. On the other hand, we would like to estimate position change over time. This is a **trade-off**

For example, if the **political debate changes and new vocabulary** enters the political lexicon in election *t*, then this will differentiate the texts at point *t* from those at point *t-1*

In fact, in this instance, we are likely to pick up a **policy agenda** shift in texts, whereas we are interested in party position change

Dynamic Estimation

Potential route to addressing this issue: carefully select the **words** that enter the analysis!!!

Thus, if there is movement of parties, it can only be due to **different word usage**

This requires that the **word data over time** must be comparable at a minimum level

Dynamic Estimation

Take as an example the set of parties' manifestos in Germany since 1970 to 2005. Assume that you want to analyze such documents with Wordfish

Now assume that the political lexicon in the manifestos at election time *t* contains an issue that is no longer relevant at time *t+1*, e.g. official relations with the GDR (East Germany)

If all parties make a statement with regard to the GDR at point *t* but not at *t+1*, then the words **will not only distinguish** parties at point *t*, but also **distinguish the elections**

As a result, if all words are counted, even the rare ones, the parties are more likely to be **clustered by election**

Dynamic Estimation

The same is true if we have some changes in the **actual meaning** of some political words

Which word inclusion criteria then?

Two (main) options

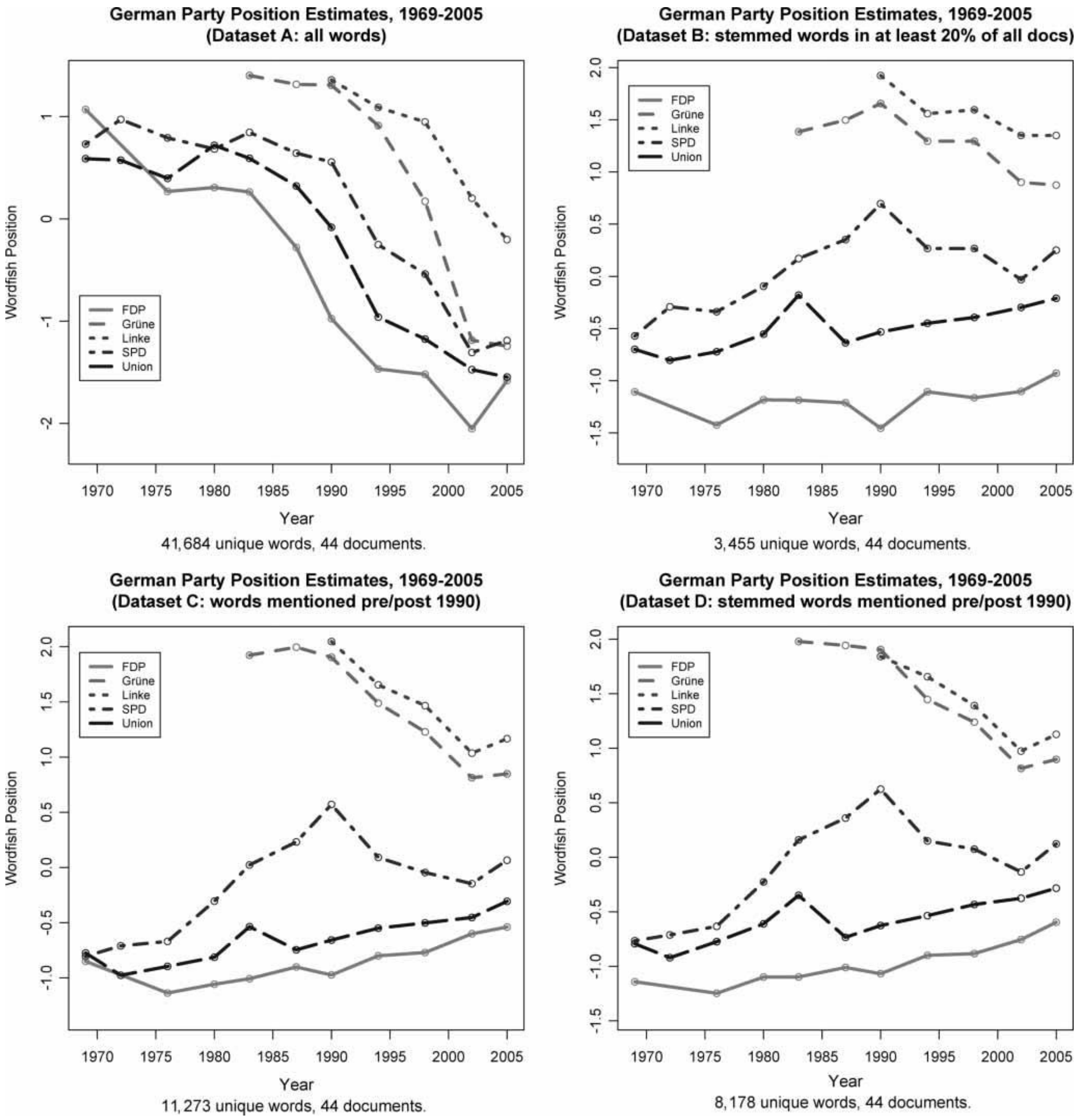
Dynamic Estimation

First alternative (non-informative priors): in the term-document matrix includes words that are **mentioned in a minimum number of documents** (say, in at least 20%), thus essentially keeping words that are deemed important enough to be mentioned either over time by one party or by several parties

Dynamic Estimation

Second alternative (informative priors): in the term-document matrix includes **only those words that appear both pre- and post-1990**, i.e., reunification added words to the German political lexicon that were not in it previously. Likewise, some words that were previously important likely fell out of use.

If we do not control for this fact, we would see a **large jump** in all parties around 1990 as they all shift their word usage to account for new political realities



German Party Position Estimates, 1969-2005
(Dataset A: all words)

41,684 unique words, 44 documents.

German Party Position Estimates, 1969-2005
(Dataset B: stemmed words in at least 20% of all docs)

3,455 unique words, 44 documents.

German Party Position Estimates, 1969-2005
(Dataset C: words mentioned pre/post 1990)

11,273 unique words, 44 documents.

German Party Position Estimates, 1969-2005
(Dataset D: stemmed words mentioned pre/post 1990)

8,178 unique words, 44 documents.

Dynamic Estimation

As suspected, **agenda effects over time** dominate the results when all words are used

Excluding **words that are specific to a given time-period** induces stability and the results are corroborated by their good face validity

An addendum about C.I.

Wordfish in the Quanteda package implements asymptotic standard errors. These SEs rely however heavily on the model being correctly specified

As a way of obtaining uncertainty estimates with weaker assumptions, Lowe and Benoit (2013) also introduced a **bootstrap procedure**, that basically iterates across different (bootstrapped) samples of the original DfM and then average the results

The Quanteda package supplies functionality for random sampling of Words [`dfm_sample`], which can be used to implement the above bootstrap procedure with relative ease

An addendum about C.I.

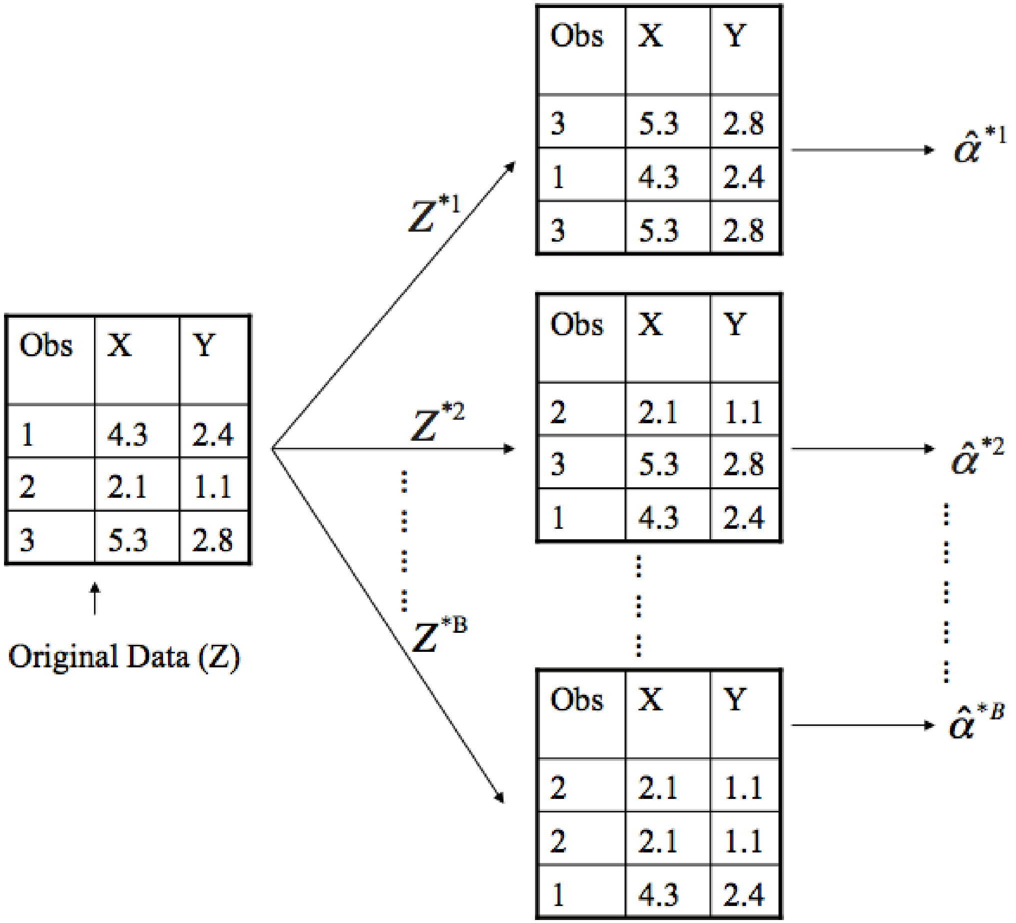
What do we mean by **bootstrapping**?

In essence bootstrapping **repeatedly draws independent samples** from our data set to create bootstrap data sets. This sample is performed with *replacement*, which means that the same observation can be sampled more than once

Each bootstrap is the used to compute the estimated statistic we are interested in (i.e., a mean or anything else – as the thetas of a Wordfish model!)

An addendum about C.I.

An example with 3 resamples



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Original Data (Z)

Z^{*1}

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

$\hat{\alpha}^{*1}$

Z^{*2}

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

$\hat{\alpha}^{*2}$

Z^{*B}

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

$\hat{\alpha}^{*B}$

An addendum about C.I.

Bootstrapping is an extremely powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method

We can in fact use all the bootstrapped data sets to compute the standard error of the desired statistics, or their 95% confidence intervals, etc.

This computation will be robust to (i.e., less affected from) sample specific characteristics

If you are interested about an example that implements this procedure, drop me an email!

Further unsupervised scaling algorithms

Suppose you want to scale legislative speeches to infer the position of legislators via an unsupervised scaling method

If we confine the analysis to speeches on a **single legislative act**, such as a motion of confidence or during the general policy speech of the PM (the approach we saw in the Japanese analysis of legislative speeches via Wordfish), no big problem!

This approach (by assumption) holds **topical variation constant** (nice thing!)

Note however that the resulting estimates are confined to the set of legislators **who spoke** and **the topic** on which they spoke

Further unsupervised scaling algorithms

But suppose now that we want to estimate the positions of MPs by analyzing all the speeches they gave across different legislative debates

In this case, of course, **topical mixes vary enormously** at the level of individual speakers, so that aggregating all the speeches across many topics by MPs and then applying a single Wordfish analysis to them wouldn’t make much sense

How to deal with that?

Wordshoal algorithm(Lauderdale and Herzog 2016): a “shoal” is a group of fish, not traveling in the same direction!

Further unsupervised scaling algorithms

Wordshoal is based on 2 stages:

The first stage uses Wordfish to scale word use variation in **each debate separately**. By doing that, we estimate the **topic-specific positions** of MPs

In the second stage, it uses **Bayesian factor analysis** to construct a **common scale** from the debate specific positions estimated in the first stage, i.e., it unifies the multiple topic-specific positions by applying factor analysis to the topic-specific positions estimated in the first stage

Further unsupervised scaling algorithms

Essentially, this allows the model to select out those debate-specific dimensions that **reflect a common dimension**, while down-weighting the contribution of those debates where the word usage variation across individuals seems to be idiosyncratic

This framework can be eventually extended to a 2-dimensional framework

Wordshoal is therefore **attractive everytime** you want to analyze several different speeches/documents per-speaker/actor taken in very different contexts (over possible different topics)

Further unsupervised scaling algorithms

Lauderdale, Benjamin E., and Alexander Herzog (2016). Measuring Political Positions from Legislative Speech, *Political Analysis* (2016) 24:374–394

To install Wordshoal:

```
devtools::install_github("kbenoit/wordshoal")
```

Quanteda command: `textmodel_wordshoal`

See an example here:

https://www.dropbox.com/s/3nc8j1yaq3p0mb1/Lab%202%202020%20Wordshoal%20example.R?dl=0

N.B. the Quanteda command allows you to estimate only a 1-dimensional world. If you are interested to estimate a 2-dimensional world, write me!

Before our second Lab

```
devtools::install_github("quanteda/quanteda.textmodels")
install.packages("cowplot", repos='http://cran.us.r-project.org')
install.packages("psych", repos='http://cran.us.r-project.org')
install.packages("PerformanceAnalytics", repos='http://cran.us.r-project.org')
install.packages("stringr", repos='http://cran.us.r-project.org')
```