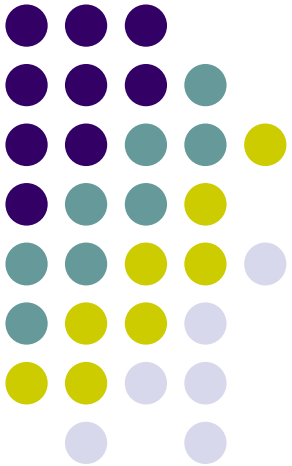
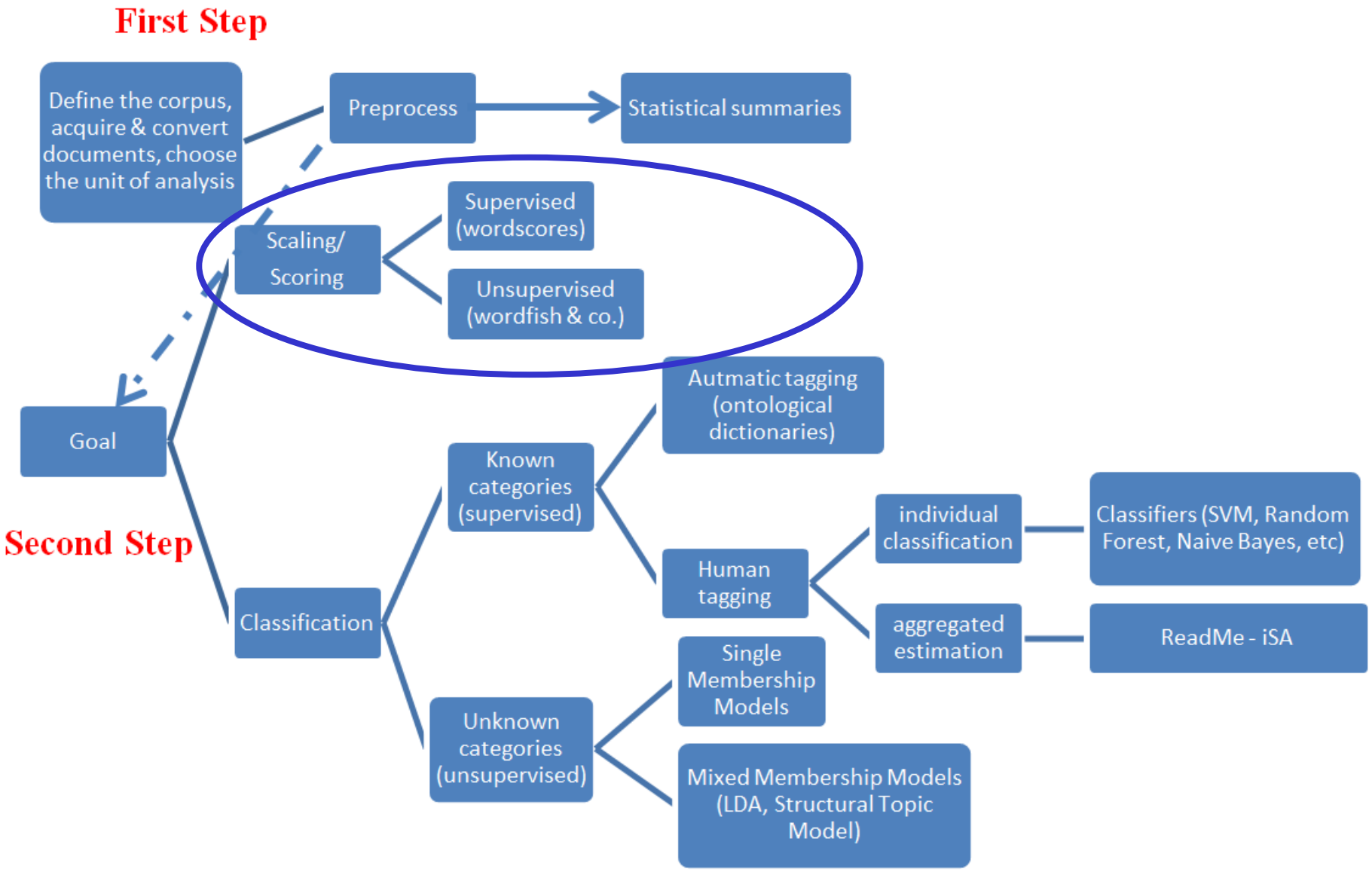
Big Data Analytics

Lecture 2

Unsupervised scaling algorithms



Our Course Map



References (unsupervised)

- ✓ Proksch, Sven-Oliver, 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-Oliver, 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. 2020. 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*, 55(2), 184-201

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 used**, not so much **WHAT they might represent**

This is something you can retrieve for example from your corpus via the `kwic` command

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!

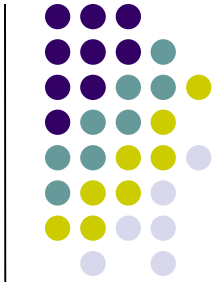
Latent models

Important theories about political and social actors concern qualities that are **unobservable through direct means**

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

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

Scaling methods



When we deal with a corpus of texts, we can view the **choice of words as the observed outcome**

As long as **certain statements are associated with particular preferences**, we can therefore use them to discriminate between positions as expressed in different texts

Accordingly, the goal of **scaling methods** is to use **some observed set of outcomes** (words in our case) to draw inferences about an actor's unobservable position on a **latent dimension(s)** *relative* to other actors

Position is here to be understood as a **preference on that dimension**

Scaling methods

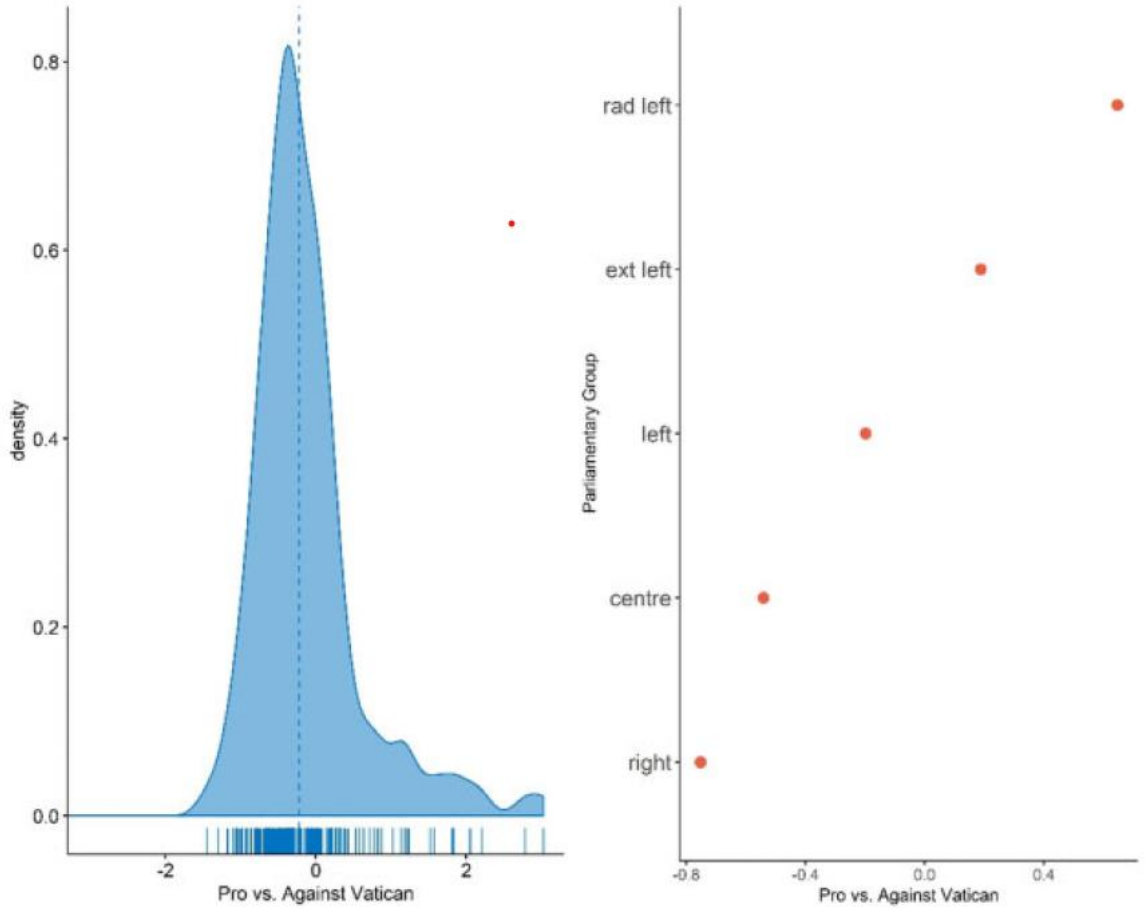
In other words, the use of a particular (set of) word(s) can provide us with **revealed preferences by the authors of the texts**

Such preferences, as we will discuss, could be related to ideology, or to some other policy (or non-policy) space

A big advantage: nearly all actors speak (or write)! And they used to speak (or write) also **long time ago**…

Scaling methods

An example: Italian MPs positions between 1861 and 1870 along a pro vs against Vatican latent dimension (Casiraghi and Curini 2021)



Types of scaling

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

What's the main difference? **Supervised Scaling Methods** (but this is true for all supervised methods!) require some kind of **a-priori information** by the researcher to produce estimates

Unsupervised Scaling Methods (but this is true for all unsupervised methods!) do not require that!

Let's start with the latter ones

Wordfish

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

How to do that?

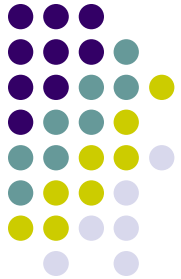
Let's introduce **Wordfish**!

Wordfish

Wordfish assumes that **relative word usage** within documents conveys information about their positions in some latent space

To give an example, this algorithm 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 latent dimension, and **discriminate** between the parties

Wordfish Estimation Process

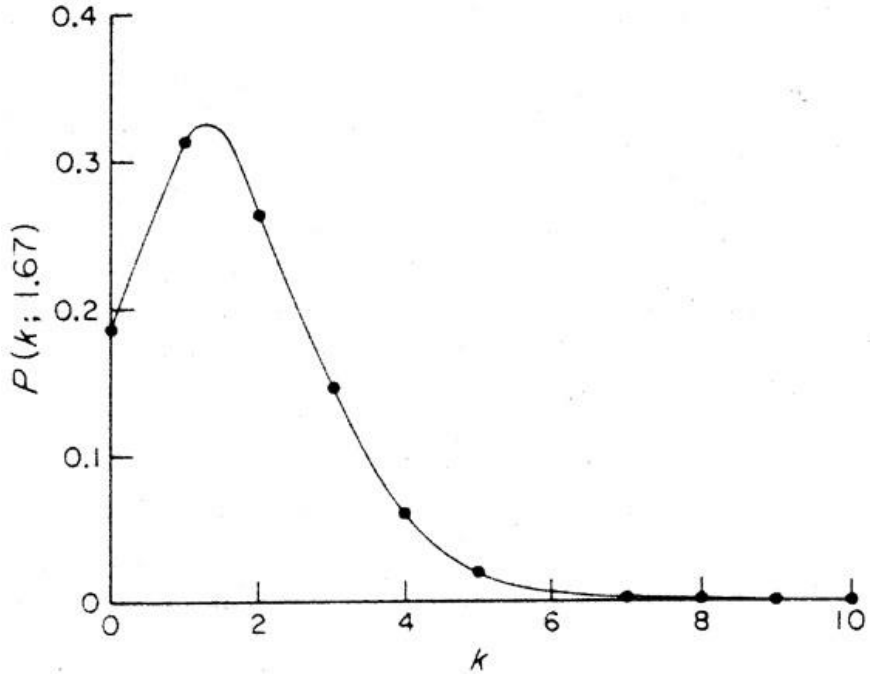


The *discover* of words that distinguish locations on a latent spectrum is made possible by adopting some statistical assumptions on the **distribution of words** employed in texts

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:

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

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

Wordfish Estimation Process

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

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

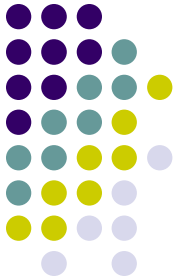
Wordfish Estimation Process

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

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 meaning, such as conjunctions or definite and indefinite articles

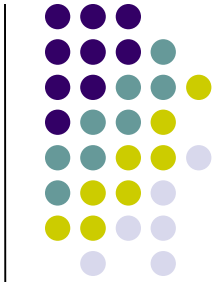
Wordfish Estimation Process



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

The **word discrimination parameters** allow the researcher to analyze **which words differentiate documents positions**

Wordfish Estimation Process



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

Finally, and *crucially*, the **document positions parameters** tells us the positions of each document relative to the other documents in the recovered latent space

This allows the researcher to estimate document positions and uncover the variations in language that are responsible for placing documents on this latent dimension

Wordfish

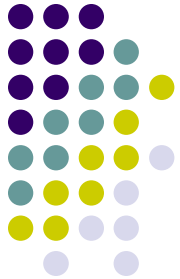
Note one important aspect: the substantial interpretation of the **estimated latent dimension** in Wordfish is completely left to the researcher

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!)

That is, unsupervised scaling methods do not require a-priori information, but they do require a lot of a-posteriori analysis!

Wordfish Estimation Process



Let's see an example

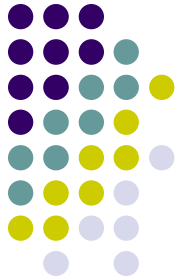
In Curini et al. (2018), we have selected all the speeches in which Japanese Prime Ministers make a general policy speech (*shoshin hyoumei enzetsu*) in the following situations:

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

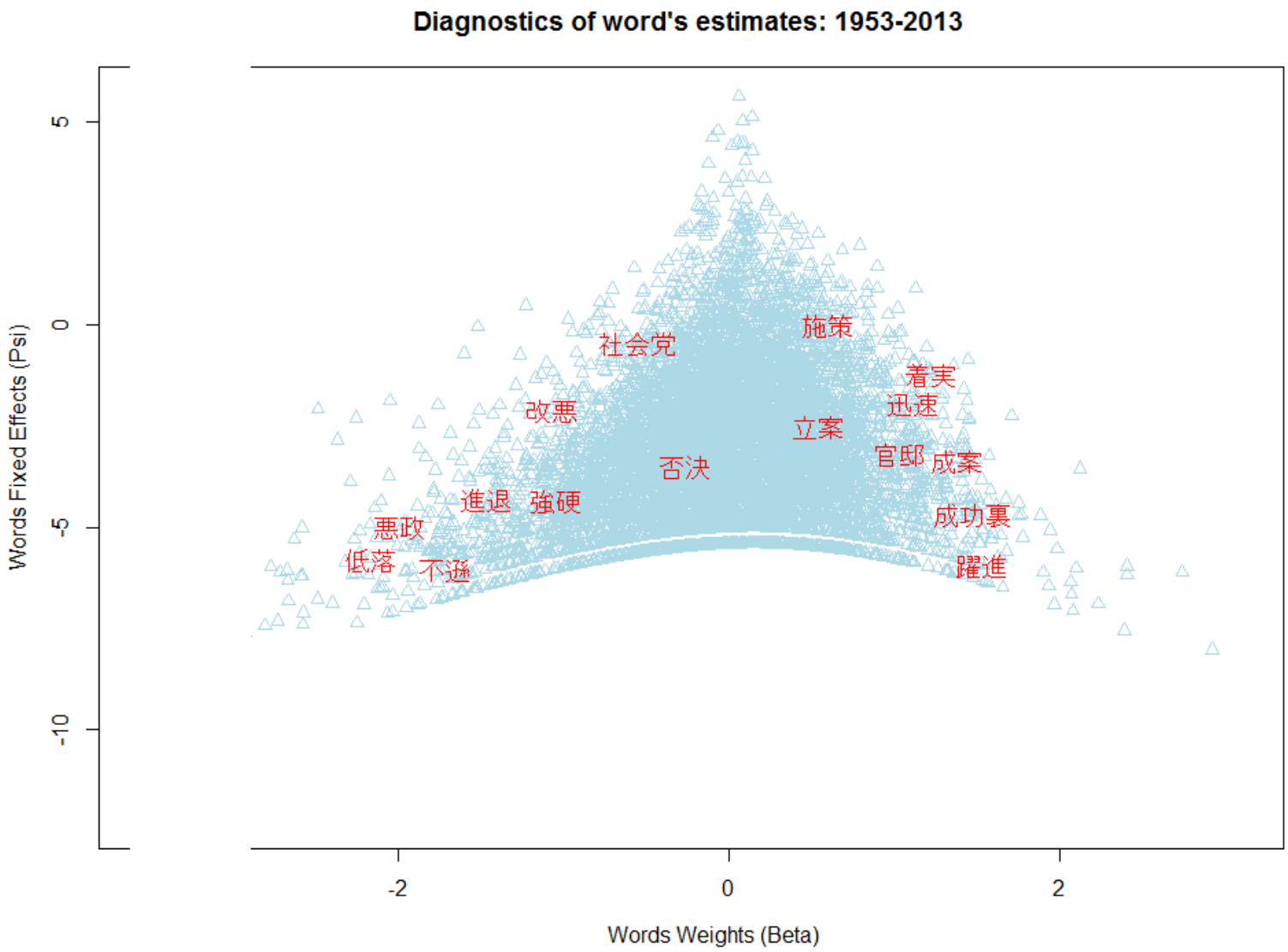
URL to get access to Japanese legislative speeches:

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



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 to infer the substantial content of the latent dimension along which the documents are going to be scaled

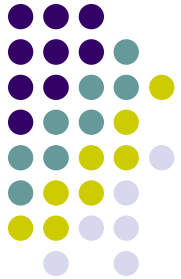
The discriminating words

In the example just saw, *bills passed* has a large *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 large 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



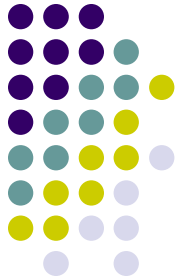
WORDFISH uses an **expectation maximization (EM) algorithm** to retrieve maximum likelihood estimates for all parameters

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

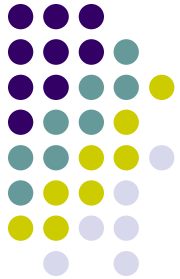
This process is **repeated until the parameter estimates** reach an acceptable level of convergence

Some challenges of doing unsupervised scaling



1. A-priori assumptions (to be satisfied) to meaningfully scale a corpus
2. Document selection
3. Dynamic estimation

A-priori assumptions



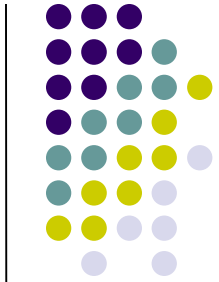
First, if the costs to articulate a position are high, authors might choose not to articulate the position for *strategic reasons*

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

A-priori assumptions

A-priori assumptions

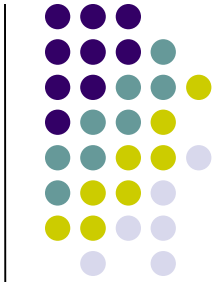


Second, the documents **should be informative about the differences** we seek to estimate

Particularly in contexts where there are **strong common norms about how to phrase a documen**t (as with highly technical legislative or legal documents), it can be difficult to scale documents

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

Document Selection



Document selection is essential and possibly the most tricky task in the estimation process

Wordfish estimates a **single dimension**, and the information contained in this **dimension depends only upon the texts** that the researcher chooses to analyze (w/o any a-priori human contribution)

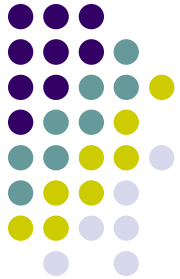
Therefore, the **selection of texts should depend** on the particular dimension the researcher would wish to examine

Document Selection

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, then the analysis should perhaps be conducted using all parts of an election manifesto, for example, assuming that such documents are encyclopedic statements of policy positions

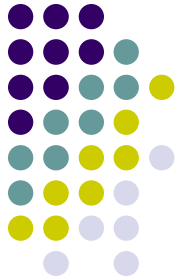
Document Selection



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 Selection



However, **never take for granted** the content of the content of the dimension you are extracting without careful validation. You need always to validate it! Remember the fourth principle!

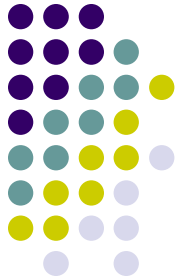
In the previous example about Japan, we actually capture an opposition-cabinet latent dimension not for example an ideological dimension!

Document Selection

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

Therefore be careful when you mix texts dealing with completely different topics, while using Wordfish on them! Why?

Document Selection



Wordfish will recognize differences in word use between texts as indicative of their different 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 with Wordfish, and if they are, it will often result in the main latent dimension being grossly miss-specified

Document Selection

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?

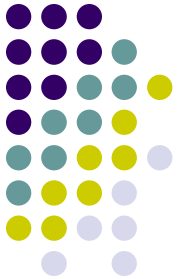
Document Selection

But suppose that you still want to extract one single latent dimension able to differentiate the authors of texts covering different topics (or themes)

For example: you want to analyze the positions of Mps during legislative debates discussing foreign-policy with respect to different countries (Venezuela in one debate and Iraq in another one, so that in the first debate several terms related to Venezuela are employed; while in the second debate several terms related to Iraq are employed)

If you employ Wordfish in this case, you will obtain once again a latent scale that will differentiate between texts discussing about Venezuela on one extreme of the latent dimension and texts discussing about Iraq on the other extreme

Document Selection



So how to deal with that?

First option: carefully select the **words** that enter the analysis, so that the **word data across debates** can be comparable

Thus, if parties are located in a different position along the latent dimension, it can only be due to **different word usage** starting from a comparable set of words

In our case, that would imply for example deleting before the analysis any specific words related to Venezuela and Iraq (via `token_remove`)

Document Selection

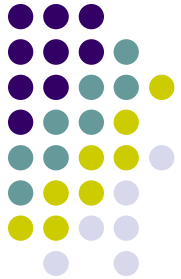
So how to deal with that?

Second option: change the algorithm!

Think about this more challenging example: suppose that we want to estimate the positions of MPs along some common latent dimension 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 (in a much higher way than in the previous case where at least all speeches were covering foreign policy…), 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

Document Selection



How to deal with that?

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

Document Selection

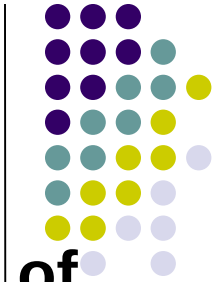
Wordshoal is based on 2 stages:

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

What do you mean by that?

Document Selection



Any factor analysis (FA) is used to **reduce the number of dimensions** within a data set **by choosing** only those "factors" (1 or more) that account for **most of the variation in the original multivariate data** and to summarize the data with little loss of information by projecting them onto a lower dimensional subspace

This can result in a **good approximation** of the original data

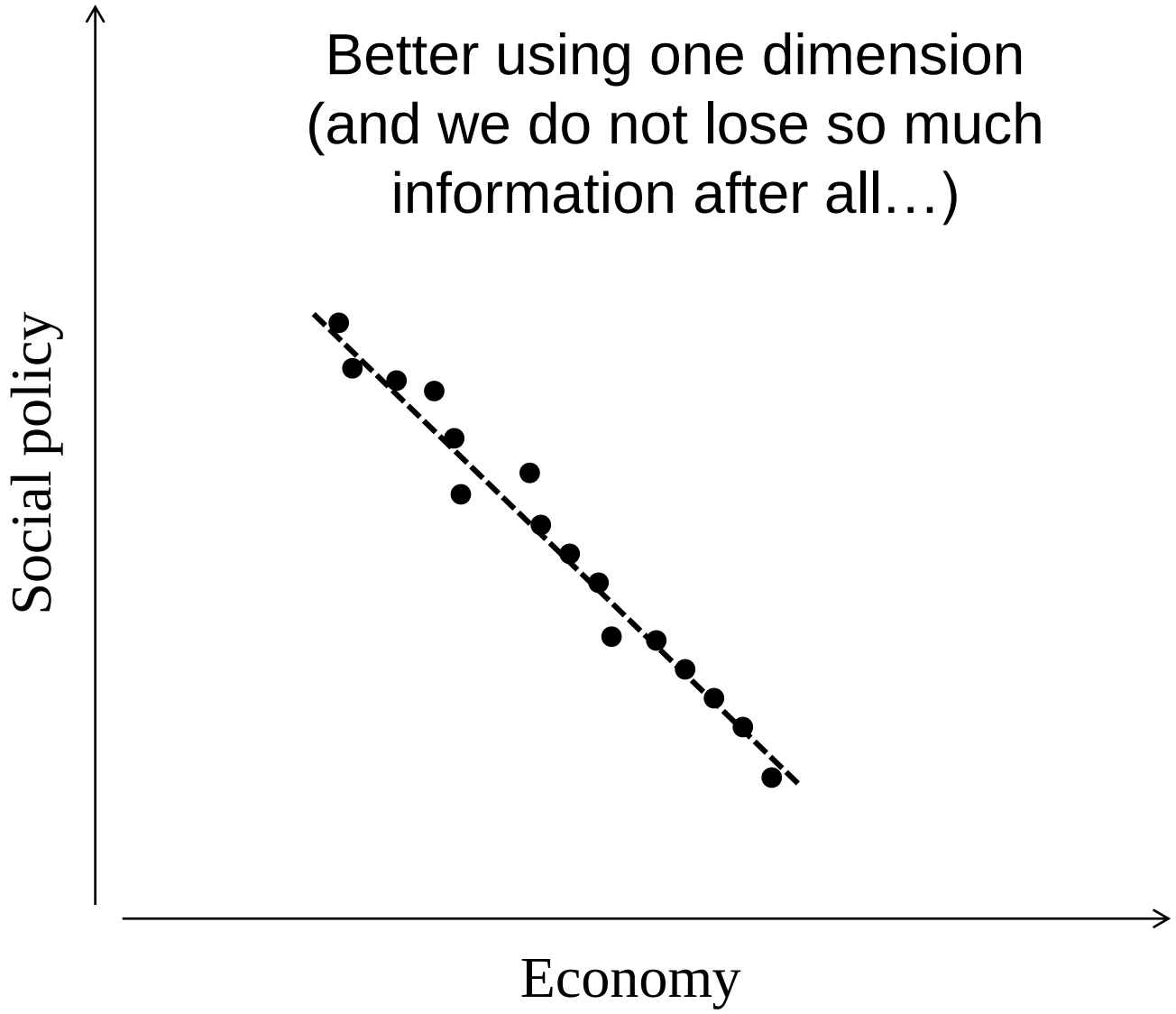
FA is especially useful when there is a **high-degree of correlation** present in our dataset (if correlation in your data is zero, there is no way to do any reasonable dimension reduction!!!)

Document Selection

Do we really need two dimensions to capture all the variance in party positions?

Document Selection

Better using one dimension
(and we do not lose so much
information after all…)



Document Selection

In our case, the (Bayesian) FA allows 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

Why a Bayesian FA? Its advantage is that it allows missing values in observed values. Going back to our example, this property is desirable if each MP does not necessarily speak in all legislative debates; thus, considerable debate-specific positions may be unobservable

Document Selection

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) – as long of course there is *some correlation* about authors' positions across the contexts…

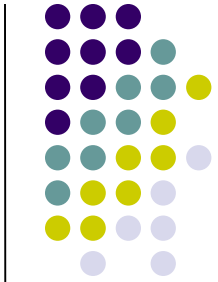
Lauderdale, Benjamin E., and Alexander Herzog (2016). Measuring Political Positions from Legislative Speech, *Political Analysis* (2016) 24:374–394

To install Wordshoal:

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

Quanteda command: `textmodel_wordshoal`

Document Selection

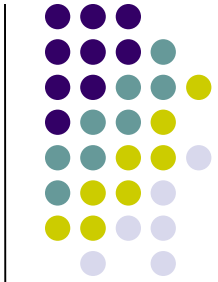


Finally, Wordfish is data-hungry!

According to Egerod and Klemmensen (2020), scaling corpora with fewer than approximately 1,800 words in the average text is infeasible using Wordfish

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

Dynamic Estimation



Using texts to estimate policiy 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

But there is a **trade-off** here to be considered…

Dynamic Estimation

For example, if **public debate changes and new vocabulary** enters the public lexicon at time *t*, then this fact per-se (i.e., the change in the vocabulary) will differentiate texts at point *t* from those at point *t-1* irrespective (or above of) any “true” change in the authors’ positions along the same latent dimension!

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

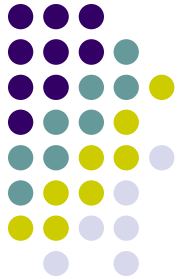
Now assume that the political lexicon in the manifestos at election time *t* contains an issue (and a vocabulary) that is no longer relevant at time *t+1*, e.g. official relations with the GDR (East Germany)

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**


German Party Position Estimates, 1969-2005
(Dataset A: all words)
Wordfish Position
1
0
-1
-2
FDP
Grüne
Linke
SPD
Union
1970
1975
1980
1985
1990
1995
2000
2005
Year
41,684 unique words, 44 documents.

Dynamic Estimation



Which are the potential route to addressing this issue?

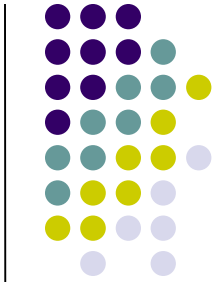
Once again, we must carefully select the **words** that enter the analysis by creating a set of word data that can be comparable at a minimum level

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

Which word inclusion criteria then?

Two (main) options

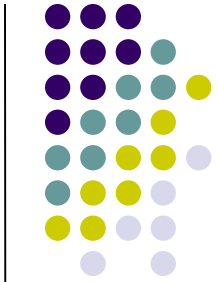
Dynamic Estimation



First alternative (non-informative priors):

- ✓ in the DfM 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

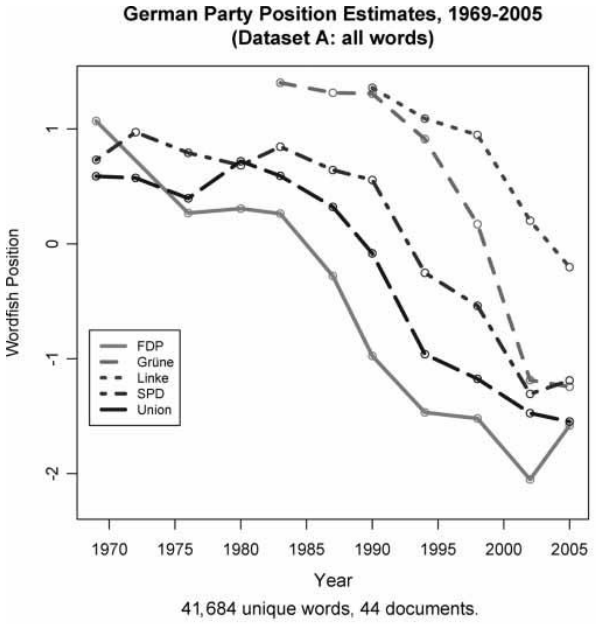


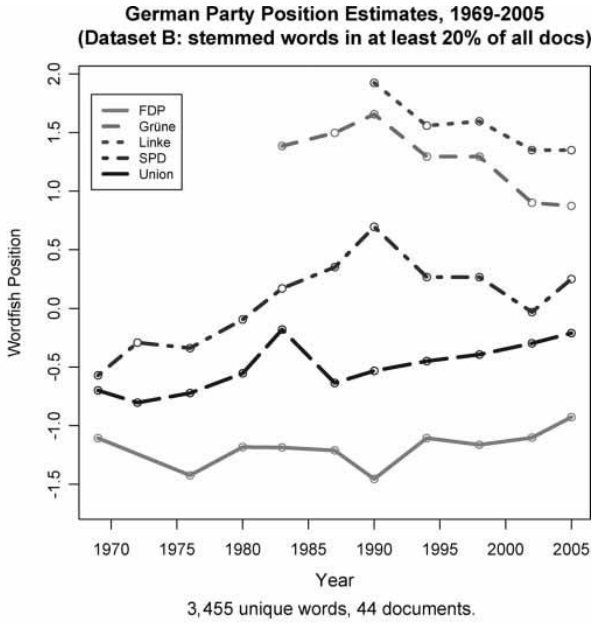
Second alternative (informative priors)

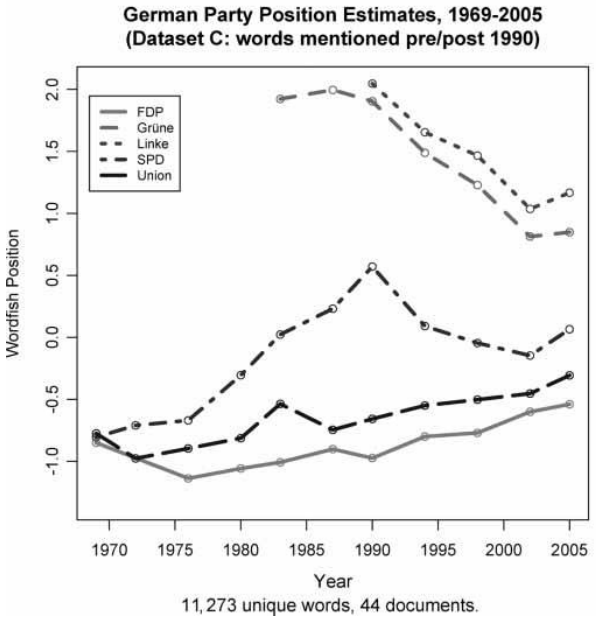
- ✓ in the DfM includes **only those words that appear both pre- and post-1990**, i.e., reunification added words to the German political lexicon that were not in it previously. Likewise, some words that were previously important likely fell out of use

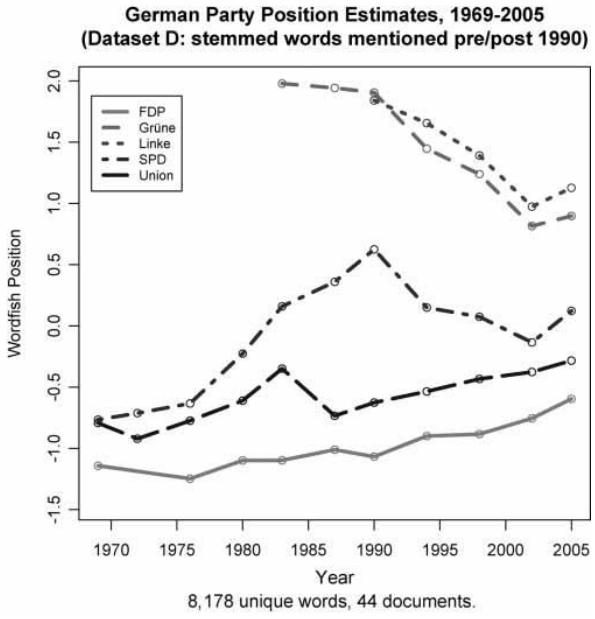
If we do not control for this fact, we would see a **large jump** in all parties around 1990 as they all shift their word usage to account for new political realities

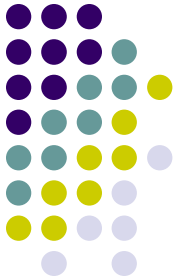
And indeed…


German Party Position Estimates, 1969-2005
(Dataset A: all words)
Wordfish Position
Year
FDP
Grüne
Linke
SPD
Union
41,684 unique words, 44 documents.


German Party Position Estimates, 1969-2005
(Dataset B: stemmed words in at least 20% of all docs)
Wordfish Position
Year
FDP
Grüne
Linke
SPD
Union
3,455 unique words, 44 documents.

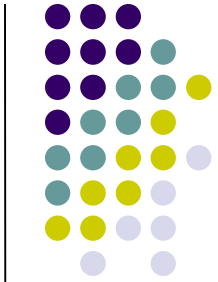

German Party Position Estimates, 1969-2005
(Dataset C: words mentioned pre/post 1990)
Wordfish Position
Year
FDP
Grüne
Linke
SPD
Union
11,273 unique words, 44 documents.


German Party Position Estimates, 1969-2005
(Dataset D: stemmed words mentioned pre/post 1990)
Wordfish Position
Year
FDP
Grüne
Linke
SPD
Union
8,178 unique words, 44 documents.



IMPORTANT!!!

Before using rtweet



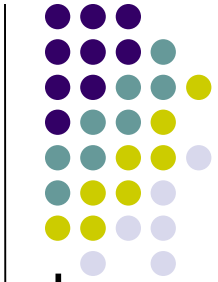
We will use tomorrow the `rtweet` package: so start to install it!

install.packages("rtweet", repos='http://cran.us.r-project.org')

install.packages("httpuv", repos='http://cran.us.r-project.org')

install.packages("ggmap", repos='http://cran.us.r-project.org')

Before using rtweet



Then open an R session and type the following commands. Plz let me know if you are able (or not) to download the 10 tweets:

```
library(rtweet)

library(httpuv)

rt <- search_tweets( "#rstats", n = 10,
include_rts = FALSE)

print(rt$text[1:10])
```

If you have any problems, plz also take a look at there:
https://cran.r-project.org/web/packages/rtweet/vignettes/auth.html

Optional

Before we can start geocoding data, we need to obtain an API key from Google. Go to the registration page, and follow the instructions (select all mapping options)

The **geocoding API** is a free service, but you nevertheless need to associate a credit card with the account.

Please note that the Google Maps API is not a free service. There is a free allowance of 40,000 calls to the geocoding API per month, and beyond that calls are $0.005 each

This implies that basically you have a monthly free limit of $200 (more than enough…)

To register you need to have: a) a gmail account; b) a credit card

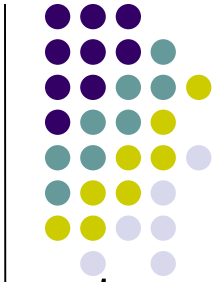
Optional

After you finish the registration (if everything hopefully works fine!) Google gives you back an API number. Save it!

Then type:

```
library(ggmap)
register_google(key = "NUMBER OF YOUR GOOGLE API!")
geocode(c("White House", "Uluru"))
You should get this result back:
# A tibble: 2 x 2
    lon   lat
  <dbl> <dbl>
1 -77.0  38.9
2 131.  -25.3
```

Optional



If you are able to get the Google API, but GGMAP does not get any results back, enable the “geocoding app” in your console developer. Check how to enable GOOGLE API here

About social media data

When dealing with social-media data, you should be always very careful about privacy:

1. what kind of information can be ethically gathered about the users (public information)
2. how published data should look like to comply with privacy regulations (like the GDPR)
3. and what consequences violating the social network's terms of service may entail for the researcher

About social media data

Some good readings about these points:

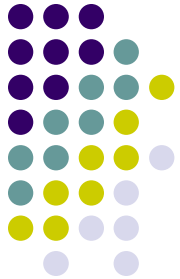
Computational Research in the Post-API Age

What You Can Scrape and What Is Right to Scrape: A Proposal for a Tool to Collect Public Facebook Data

(Supplementary Material with a great code in R to use with Facebook!)

An interesting paper with a review of nine different free-of-charge and low-cost software tools for studying Twitter: "Free and Low-Cost Twitter Research Software Tools for Social Science"

Before our second Lab



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')

install.packages("dplyr", repos='http://cran.us.r-project.org')

install.packages("gridExtra", repos='http://cran.us.r-project.org')

Before our second Lab

install.packages("maps", repos='http://cran.us.r-project.org')

install.packages("leaflet", repos='http://cran.us.r-project.org')

install.packages("rtweet", repos='http://cran.us.r-project.org')

install.packages("httpuv", repos='http://cran.us.r-project.org')

install.packages("ggmap", repos='http://cran.us.r-project.org')