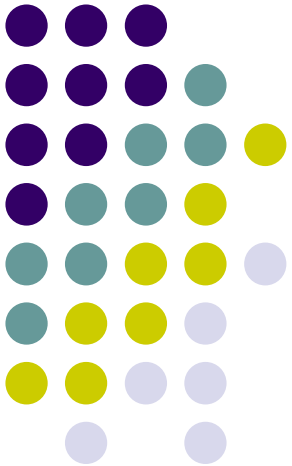
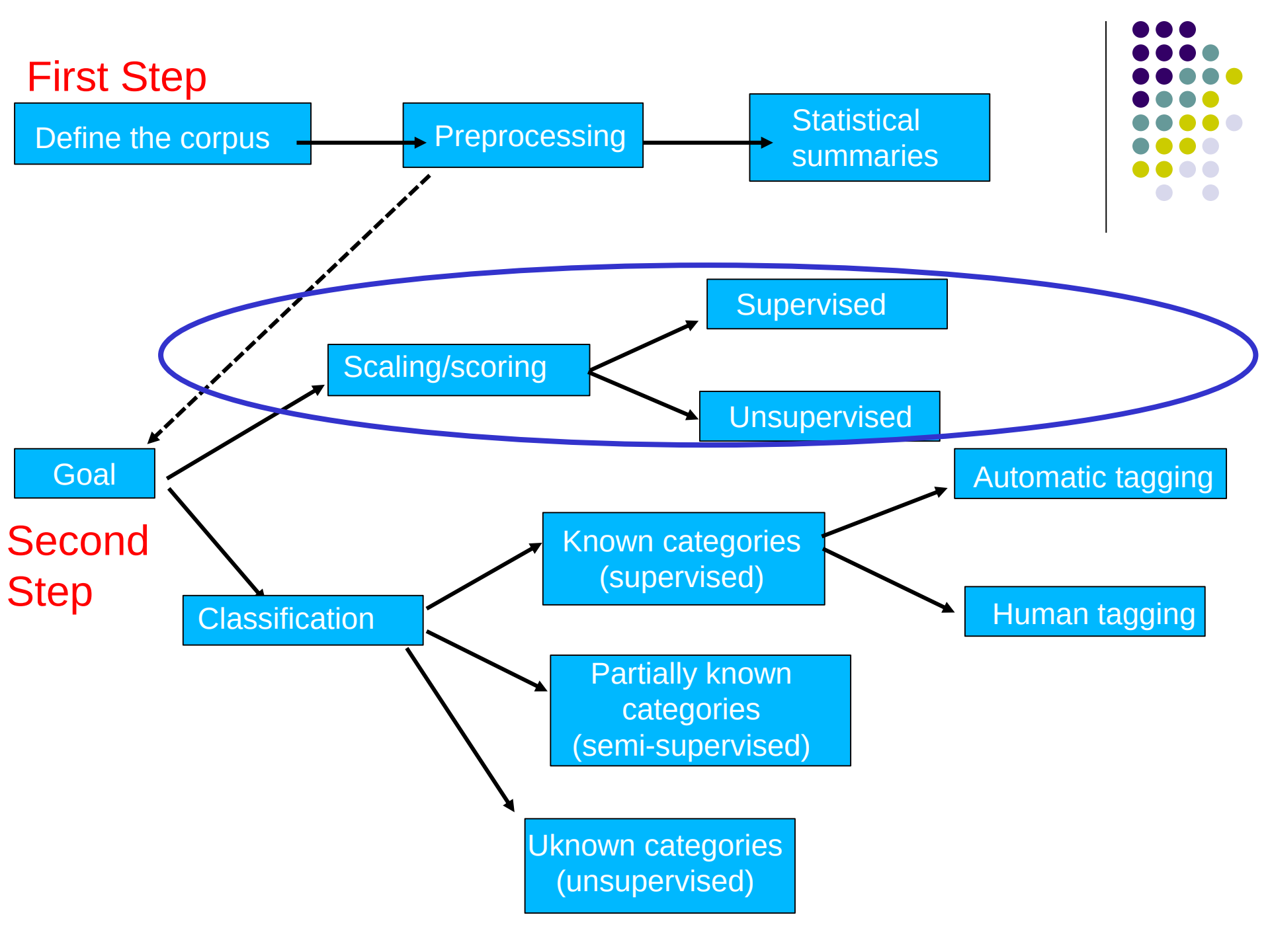
Big Data Analytics

Lecture 3

Unsupervised scaling algorithms




First Step
Define the corpus
Preprocessing
Statistical summaries
Supervised
Scaling/scoring
Unsupervised
Goal
Automatic tagging
Second Step
Known categories (supervised)
Classification
Human tagging
Partially known categories (semi-supervised)
Uknown categories (unsupervised)

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

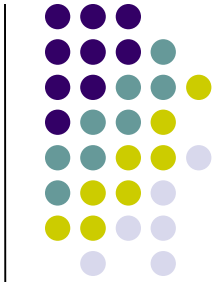
Wordfish

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**!

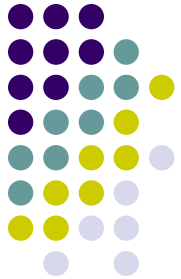
Wordfish



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

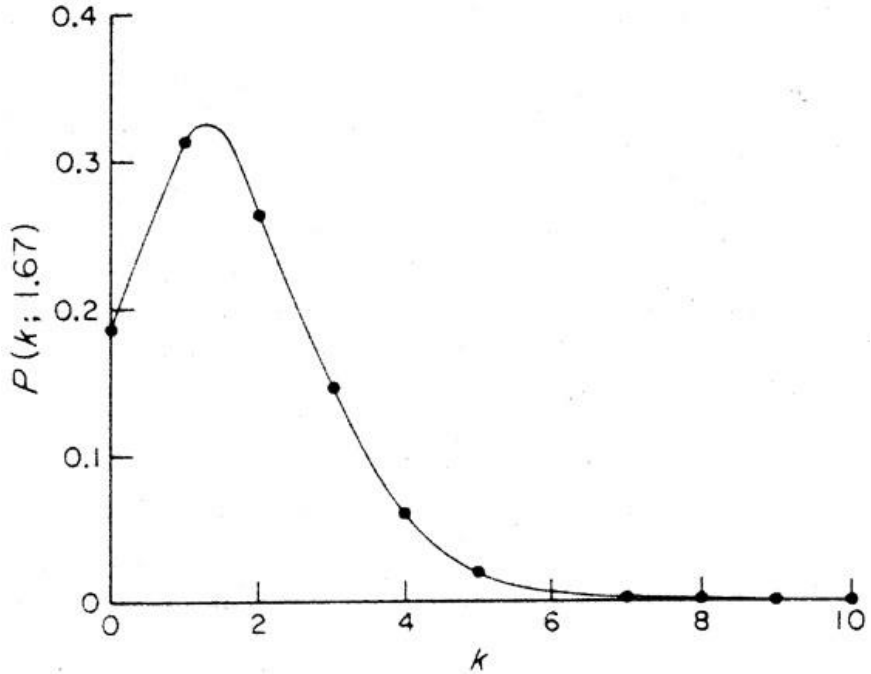


The *discover* of words that distinguish locations on a latent spectrum is made possible by adopting some statistical assumptions on the **distribution of words** employed in texts

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

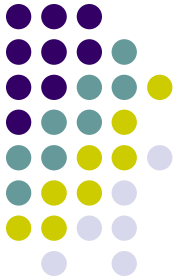
Wordfish Estimation Process

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

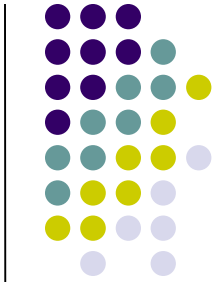
Wordfish Estimation Process



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

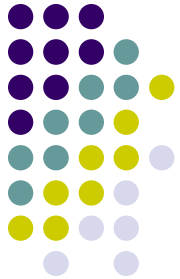
Note one important aspect: the substantial interpretation of the **estimated latent dimension** in Wordfish is completely left to the researcher

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

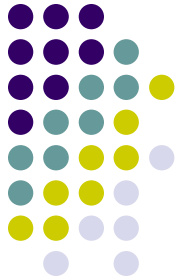
In Curini, Hino & Osaki (2020), we have selected all the speeches in which Japanese Prime Ministers make a general policy speech (*shoshin hyoumei enzetsu*) in the following situations:

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

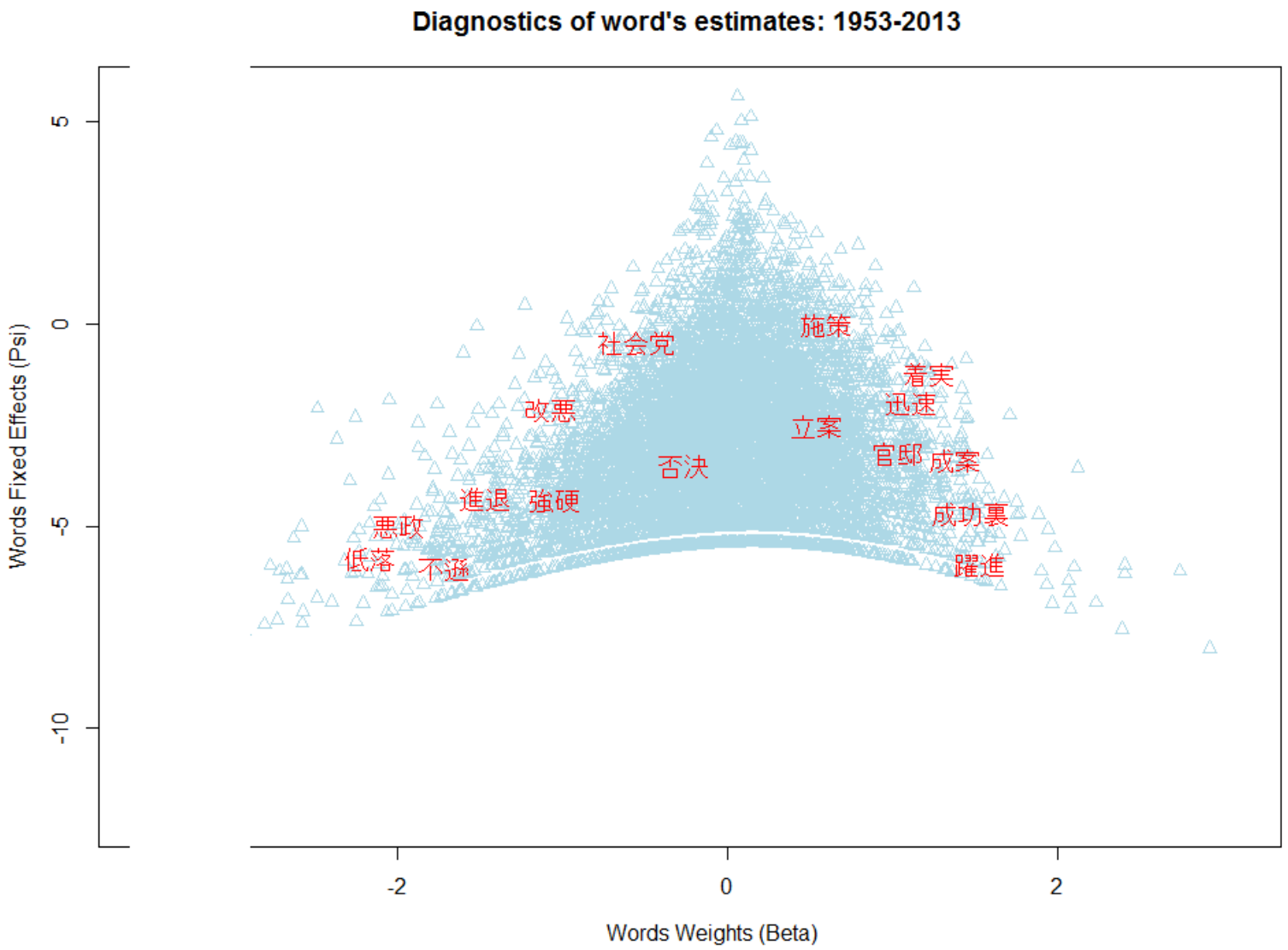
URL to get access to Japanese legislative speeches:

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

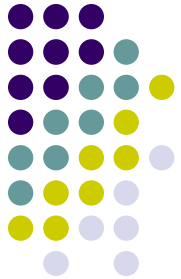
The discriminating words

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, ceteris paribus, a *positive score* along the latent dimension

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

Therefore the latent dimension is related to an *opposition-cabinet scale*?

More formally



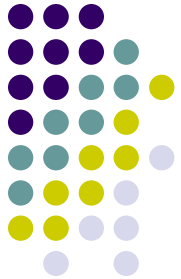
WORDFISH uses an **expectation maximization (EM) algorithm** to retrieve maximum likelihood estimates for all parameters

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 (this also applies to Wordscores…)
2. Interpretation (including c.i. estimation)
3. Document selection
4. Dynamic estimation

A-priori assumptions

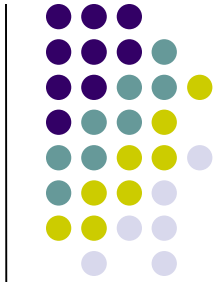
First, if the costs to articulate a position are high, authors might choose not to articulate the position for *strategic reasons*

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

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

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!

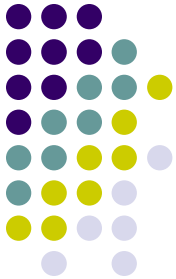
An addendum about C.I.

Wordfish in the Quanteda package implements asymptotic standard errors. These SEs rely however heavily on the model being correctly specified (an heroic assumption when dealing with text-analysis: remember the First Principles!)

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.



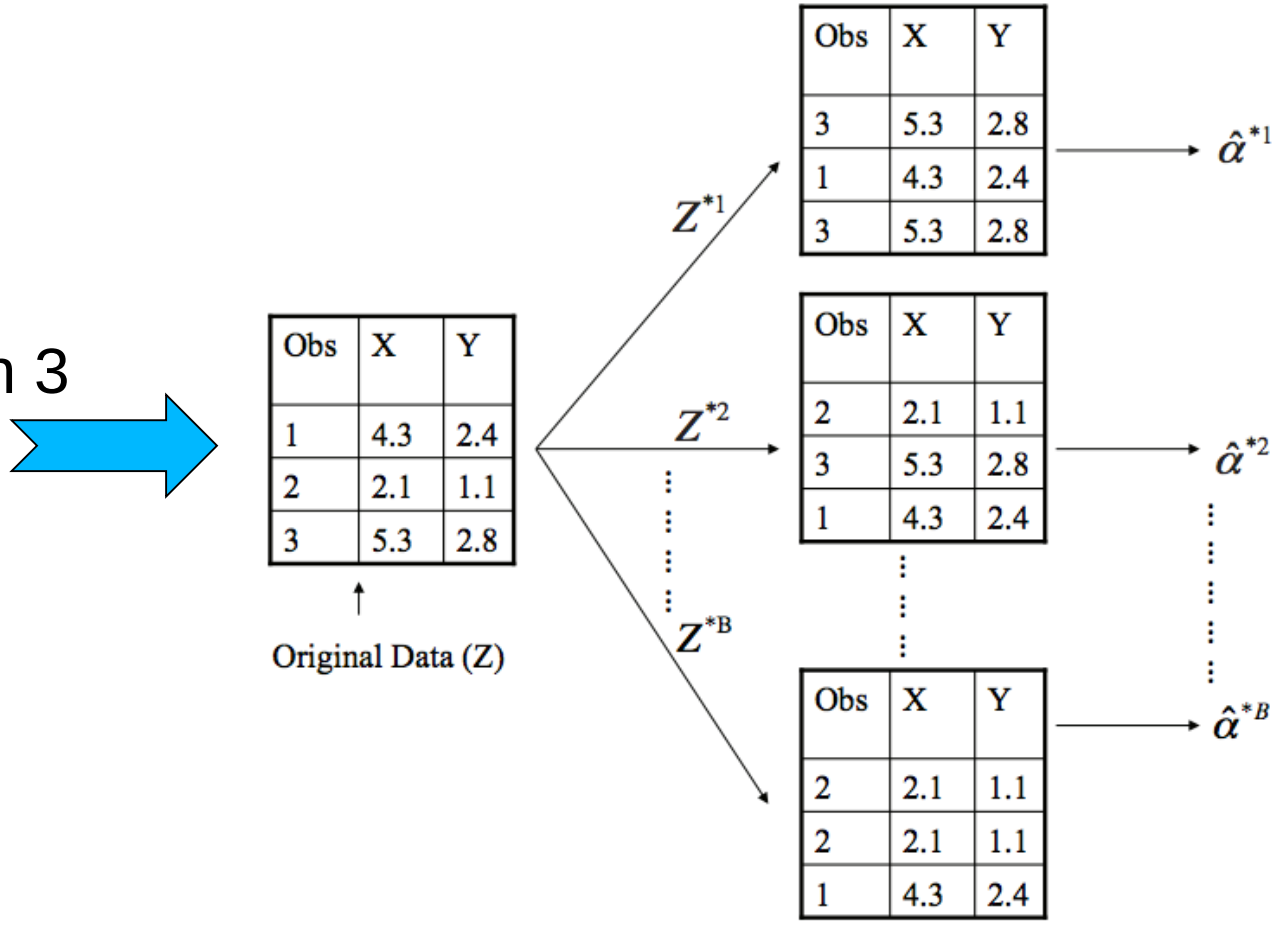
What do we mean by **bootstrapping**?

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



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

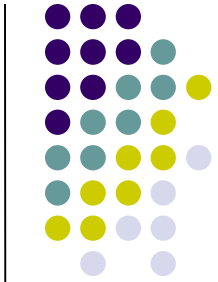
We can in fact use all the bootstrapped data sets to compute the standard error of the desired statistics, or their 95% confidence intervals, etc.

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

That is, if you are writing a paper using Wordfish, use boostrapped c.i.!!!!

If you are interested about an example that implements this procedure, take a look at this EXTRA script

Document Selection



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)

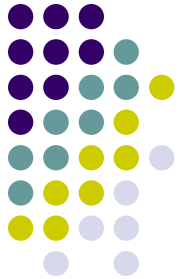
Therefore, the **selection of texts should depend** on the particular dimension the researcher would wish to examine

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

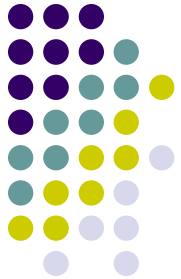
Document Selection



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**

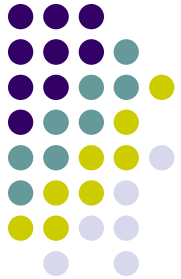
Document Selection



The estimated single latent dimension is therefore always **a function** of the selection of the text corpus

Accordingly, be careful when you mix texts dealing with completely different topics into the same corpus, 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

On the contrary, Wordscores is robust to **irrelevant text in the virgin documents**

Reference texts that contain language about two extremes of environmental policy, for instance, are unlikely to contain words about health-care

Scaling a virgin text using a model fitted to these environmental texts will therefore scale only the terms related to (and hence only the dimension of) environmental policy, even if the document being scaled contained out-of-domain text related to health care

This is an advantage with respect to Wordfish

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

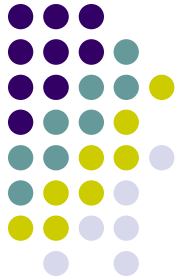
So how to deal with that?

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

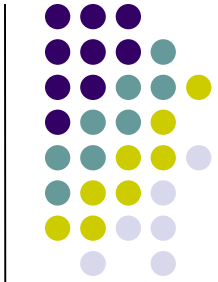


How to deal with that?

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

See the EXTRA slides on this point (if you are interested!)

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 (much more than Wordscores for example…)

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 also for Wordfish

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

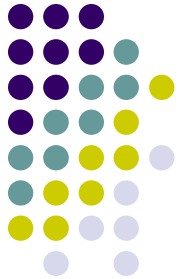
Now assume that the political lexicon in the manifestos at election time *t* contains an issue (and a vocabulary) that is no longer relevant at time *t+1*, e.g. official relations with the GDR (East Germany)

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?

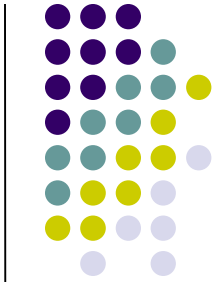
Once again, we must carefully select the **words** that enter the analysis by creating a set of word data that can be comparable at a minimum level

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

Which word inclusion criteria then?

Two (main) options

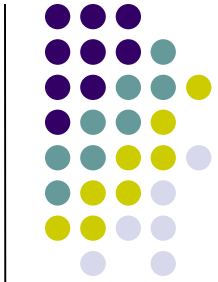
Dynamic Estimation



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

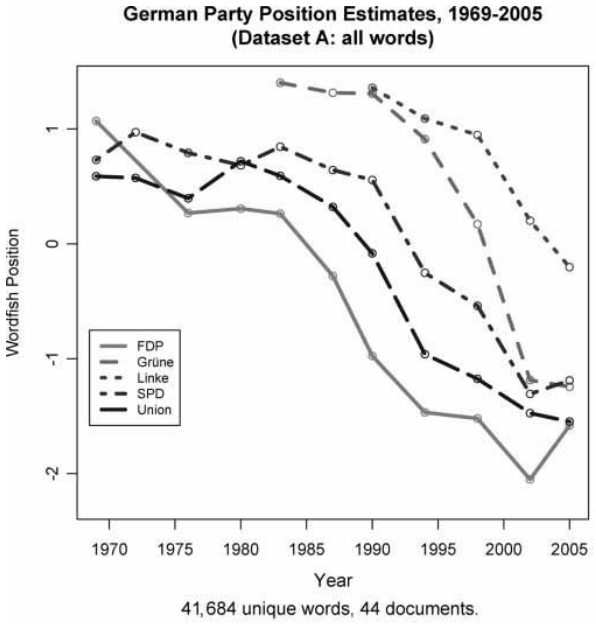


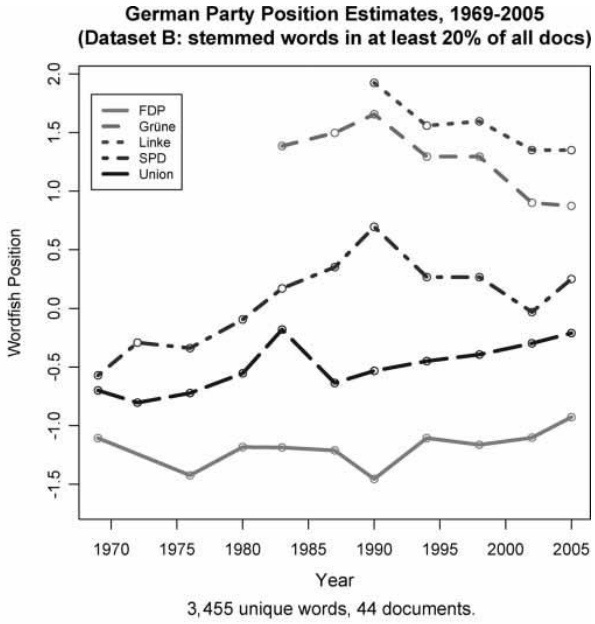
Second alternative (informative priors)

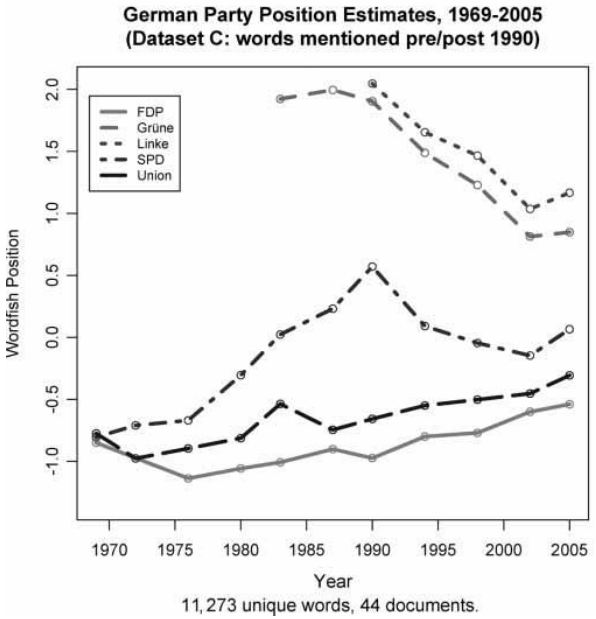
- ✓ in the DfM includes **only those words that appear both pre- and post-1990**, i.e., reunification added words to the German political lexicon that were not in it previously. Likewise, some words that were previously important likely fell out of use

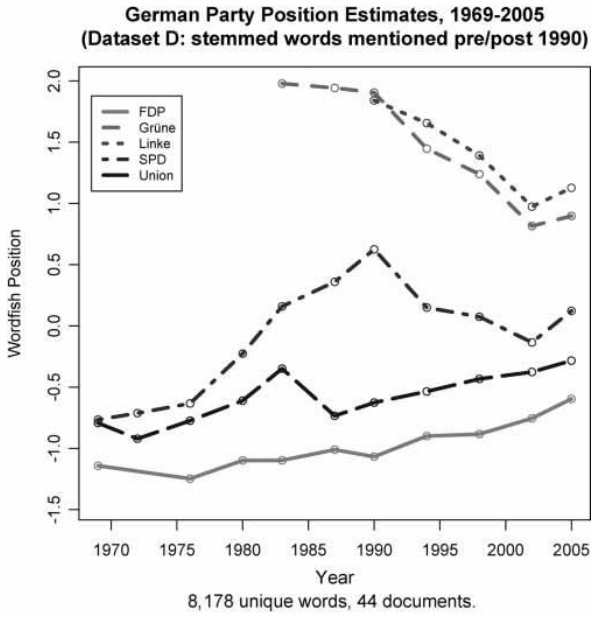
If we do not control for this fact, we would see a **large jump** in all parties around 1990 as they all shift their word usage to account for new political realities

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.

Before our second Lab

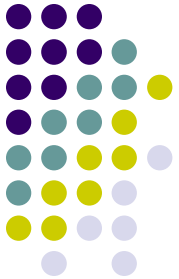
install.packages("dplyr", repos='http://cran.us.r-project.org')

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



IMPORTANT!!!

Streaming api with rtweet

Tomorrow we will search tweets via streaming api (not anymore rest api. Which is the difference? You will discover it tomorrow!)

However for doing it is much better to employ the latest version of `rtweet` (1.0.2)

So install it via: *install.packages("rtweet", repos = "http://cran.us.r-project.org")*

This will replace the previous version (0.7.0). You can however return to the previous version by re-installing it via:

devtools::install_version("rtweet", version = "0.7.0", repos = "http://cran.us.r-project.org")

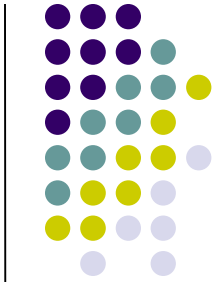
Streaming api with rtweet

Of course this is not very efficient (there are ways that allow you to employ simultaneously two different versions of the same R package) but it avoids any confusion (hopefully…)

Summing up (my suggestion):

- ✓ if you want to employ a rest-API go with the 0.7.0 package (given that it gives you access to more metadata concerning each single tweet: 90 vs. 43)
- ✓ If you want to employ a streaming-API, go with the 1.0.2. package (also because, it works only with this version!)
- ✓ Finally, if you want run a streaming-API + geo-location, it is once again better to go with the 0.7.0 package (once again more metadata…)

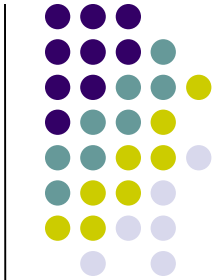
Geocoding



We will use tomorrow some geocoding tags within Twitter

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) – this is optional! You can live even w/o such API…

Geocoding



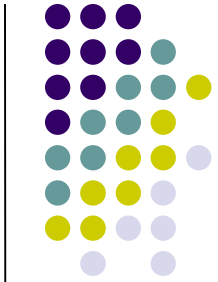
The **geocoding API** is a free service, but you nevertheless need to associate a credit card with the account.

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

Geocoding



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

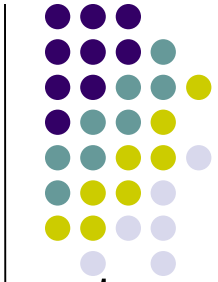
Then type:

```
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
```

Geocoding



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