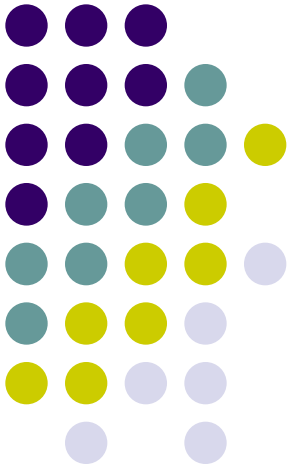
Polimetrics

Lecture 6 – Lab session
Let’s run Wordfish!



How to run WORDFISH

The WORDFISH algorithm is implemented as a function in R

The estimation involves the following steps:

(1) document processing

(2) creation of a word count dataset

(3) running Wordfish in R using the **Austin package** (if you are interested, take a look at the Quanteda package [Benoit et al. 2016] as well)

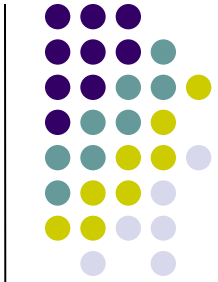
(4) diagnostics

Document Processing

Document processing is essential and possibly the most arduous task in the estimation process

First, researchers should predefine the sets of texts to be analyzed. **Second**, these texts need to be processed and all unnecessary information must be removed. Third, the **spelling** needs to be checked in all texts

Document Processing



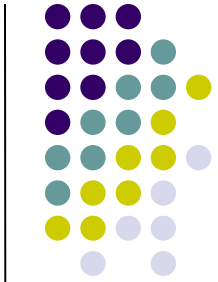
1. **Choosing documents & policy dimension**

The selection of texts will depend on what kind of policy dimension should be analyzed

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

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

Document Processing



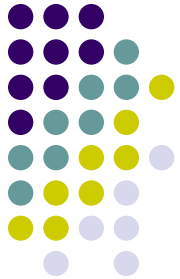
1. **Choosing documents & policy dimension**

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

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

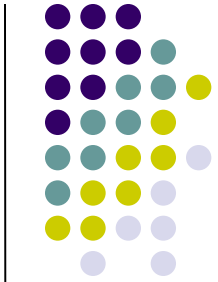
Document Processing



WORDFISH does not estimate **multiple dimensions**, only a single dimension, 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



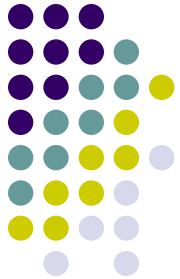
2. **Removal of unnecessary information**

Some texts include text data that researchers may prefer to remove prior to the estimation. Examples include the listing of speakers' or parties' names, self-reference of party names, headers and footers, enumeration, bullets, section headings, etc. This can either be done manually or with the help of pattern-matching using customized PERL or PYTHON scripts.

3. **Spell check**

Researchers should also ensure that the spelling of words is consistent across documents

Document Processing

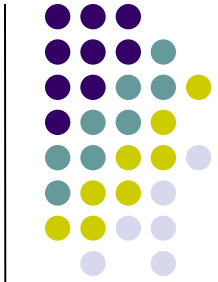


4. **Machine readable format**

Finally, once appropriate documents (or subsections of documents) have been selected, the researcher must ensure that they are in **machine readable format**

If the document is a scanned version of a manifesto, for example, converting it to a **text file** will most likely require running optical character recognition software over the documents, at which point additional error might be added to the data

Creating a term-document matrix



After document processing, a word count dataset must be generated!

This can be done using any available word count programs

Alternatively, you can use other free programs such as YOSHIKODER and JFREQ

We will use JFREQ: http://conjugateprior.org/software/jfreq/

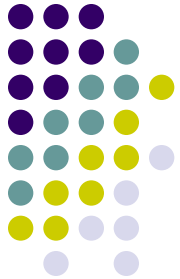
I suggest you to use the latest version of JFREQ

Which encoding for language?

http://scratchpad.wikia.com/wiki/Character_Encoding_Recommendation_for_Languages

Italian: ISO-8859-15 ; Japanese: UTF-8

Creating a term-document matrix



You could also use the R text mining package TM for that. But I would suggest you to use JFREQ (especially for Asian language)

See on my own page an example using the TM package for the Italian case

Creating a term-document matrix

Two step-procedures

1. **Stemming words**

One option when creating a term-document matrix is to count words exactly as they appear in the original documents. Another option is to **count stemmed words**

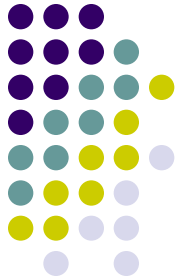
A **stemmer algorithm** removes morphological and inflexional endings from words and returns the stemmed words

For instance, the stemmer would reduce the words “fishing", “fisher", and “fished" to the root word “fish“

The advantage is that essentially **similar words** will be captured as one

Moreover, the term-document matrix will have **fewer unique words** if words are stemmed, thus making the estimation more efficient

Creating a term-document matrix



There is no clear rule whether to use a stemmer and the decision will depend on the data to be analyzed

A **potential disadvantage** is that certain compound words might be reduced to a stem thus meaning that information is lost

There is a trade-off and researchers should possibly consider **both routes** in the estimation

Moreover, instead of using words (**unigrams**) one could also use word pairs (**bigrams**), or in fact any n-gram, instead

Bigram frequencies could be scaled in the exact same fashion as those from unigrams: As a drawback, the data matrix will be larger and computation time will increase as a result

From texts to the Term-Document Matrix

WORDFISH works with any language as long as you can identify words

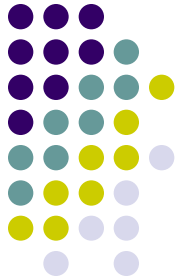
Typically, word counters identify words by white space separation

This is impossible, for instance, in Japanese, but there are work-arounds (so-called **tokenizers**, essentially a dictionary, that identify the word bounds)

Possible tokenizers: MeCab (http://mecab.sourceforge.net/)

For Japanese language: http://nomadscafe.jp/test/keitaiso/index.cgi

How tokenization works

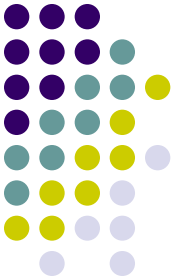


私は、日本社会党を代表して、当面する内外の諸問題につき、佐藤総理大臣にその所見をたださんとするものであります。

↓ after tokenization

私_は_、_日本_社会党_を_代表_し_て_、_当面_する 内外_の_諸_問題_に_つき_、_佐藤_総理_大臣_に_その所見_を_たださ_ん_と_する_もの_で_あり_ます_。

Creating a term-document matrix



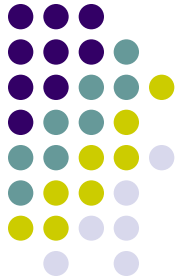
2. **Other options**

Unless there are strong reasons not to do so, it is recommended using the **lower-case option**, which transforms all words into lowercase

In addition, it is also a good idea to use the **number removal option**

Finally, very common words (**stop words** such as such as *the, is, at, which,* and *on* in English) can be removed unless there are theoretical reasons not to do so (once again, it depends on the language…not a good idea in Italian for example – personal experience)

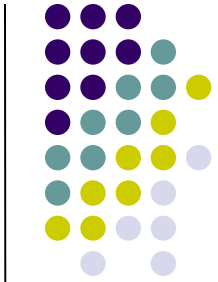
Running WORDFISH with R



We will employ the Austin package (that includes among the other things also Wordfish)

http://conjugateprior.org/software/austin/

Running WORDFISH

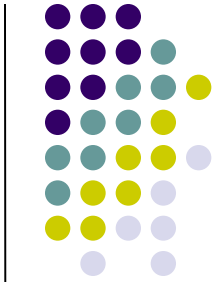


There are three steps involved in running WORDFISH: loading the data, setting WORDFISH options, and running the code until convergence has been achieved

1. **Loading the data**

If you have created the word count matrix outside of R, you should load the word count data.

Running WORDFISH



IMPORTANT:

If you use *Jfreq 2.5* you will have in the term document matrix in columns: **documents** & in rows: **words**

If you use *Jfreq 5.4 or higher* you will have in the term document matrix in **columns**: words & in **rows**: documents

In the first case just type in R: *data2 <- wfm(scores)*

In the second case just type in R: *data2 <- wfm(scores, word.margin=2)*

Where "*scores*" is the name of your term document matrix

Running WORDFISH

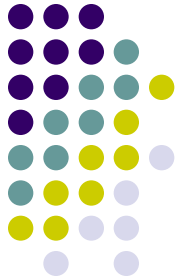
2. **Setting the options**

WORDFISH allows two identification strategies. The first one sets the mean of the positions (ω) to zero and the standard deviation to one. This is the **default identification strategy**. If you run this version, it also necessary to **indicate two documents, the first of which will have a more negative ω than the second**

This requirement ensures global identification of the model. It is recommended that you **choose documents** that you think are likely to be very different in word usage

In other words, when estimating an ideological dimension, choose texts you believe are likely represent the **opposite ends** of the political spectrum

Running WORDFISH

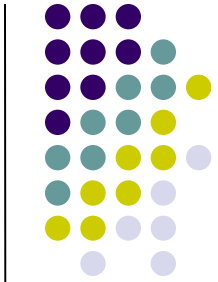


The second possibility is to simply choose two documents and assign **fixed values** to them. Then, all other positions will be estimated relative to these two anchors

The two identification strategies should produce identical results albeit on different scales

In Austin it is possible only to implement the **first identification strategy**

Running WORDFISH



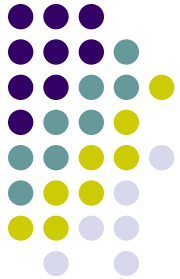
3. **Running the code**

A WORDFISH estimation can then be run simply with the following command:

results<-wordfish(data,dir=c(1,5))

In this example, *data* is the input term-document matrix and *dir* indicates which two documents (i.e. columns) are used for global identification purposes (first document to the left of the second one)

Running WORDFISH



WORDFISH output

The estimation output can easily be called from the list object results for plotting purposes or further analysis. The following output is available:

summary(results) summary of the results

plot(results) plotting document estimates ω

results$theta Document estimates, contains ω

results$alpha Document estimates, contains α

results$beta Word estimates, contains β

results$psi Word estimates, contains ψ

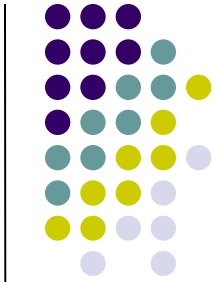
Running WORDFISH

4. **Diagnostic**

A good start for diagnostics is the analysis of word discrimination parameters.

Weights with large values mean that these words are estimated to be on the extremes of the dimension

Running WORDFISH



Let's see the Italian case: the legislative speeches by parties during the investiture debate of the Prodi cabinet (2006) [no stemmer]

Let's constrain the VER (Greens) in 2006 to have a smaller value than the FI in 2006

Replicate the Wordfish analysis on the Italian speeches by constraining the FI in 2006 to have a smaller value than the VER (Greens) in 2006

Replicate the Wordfish analysis on the Italian speeches by constraining the COM (Communist) in 2006 to have a smaller value than the FI in 2006

Re-run the example applying the "stemmer" and compare the results you got when you do not apply it