Wordfish

In recent years, computer-assisted content analysis tools became widely available for political scientists to estimate policy positions from political texts [1]. Dictionary-based analysis is one of them. ***Wordscore***, is another one (Laver, Benoit and Garry 2003). Although promising, because it is the first truly computerized technique, ***Wordscore*** still has major drawbacks that limits its usefulness in doing time-series analysis, such as constant change in political lexicon over time and the proper identification of reference text by the researcher (Slapin and Proksch 2008 : 708).

Taking a radically different approach based on sophisticated statistical modelization and heavy computerization, the ***Wordfish*** method avoids these limitations. It can produce time-series estimates, does not require the use of reference texts, has the ability to use all words in every document, and estimate the importance of each of these words. Borrowing from linguistics, it uses a naïve Bayes assumption to infer the process by which words are processed in a text. A text is represented as a vector of word counts (occurrences) and individual words are assumed to be distributed at random. The probability that each words occur in a text is independent of the position of other words in the text. While empirically false, naïve Bayes often performs well for classification (McCallum and Nigam 1998). ***Wordfish*** assume the word frequencies are generated by a Poisson process. Slapin and Proksch (2008)chose this particular distribution because of its estimation simplicity. The single paramater, λ, is both the mean and the variance. The functional form of the model is as follows:

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

Where λ_{jit} is the count of word ***j*** in party ***i'***s manifesto at time ***t***, α is a set of party-election fixed effects, ψ is a set of word fixed effects, β is an estimate of a word specific weight capturing the importance of word ***j*** in discriminating between party position, and ω is the estimate of party ***i'***s position in election year ***t***. Each platform is treated as separate party position and all positions are estimated simultaneously. That means there is no temporal constraint on the position of a party ***i'***s manifesto in election ***t***. So if a party uses words in similar relative frequencies over time, its position will remain the same. Party movement is due to changes in word frequencies, not

[1] CMP, dictionary, and *Wordscore* methods are described in the Poltext website.

to change in word signification.

The next step is to estimate the right side of the regression model by using an expectation maximization (EM) algorithm (see Dempster, Laird and Rubin 1977). The EM algorithm is an iterative procedure to compute maximum likelihood estimates for latent variables. First, the starting values are calculated. Second, the party (α and ω) parameters are estimated. Third, the word (ψ and β) parameters are estimated. Four, log-likelihood is calculated (sum of individual word likelihoods). Finally, step 2-4 are repeated until convergence.

To calculate a 95% confidence interval, a parametric bootstrap is required. The parametric bootstrap is a computer-intensive resampling method. By running the EM algorithm, λ_{jit} is calculated for each cell in the dataset that represent word frequencies. 500 new datasets are then generated, taking random draws from a Poisson distribution with parameter λ_{jit} for each cell. Using the maximum likelihood estimates as starting value, the algorithm is rerun on each of these new datasets and 500 new party positions are estimated. By using the 0,025 and 0,0975 quantiles of the simulated party positions, confidence interval is approximated. With a parametric bootstrap, confidence intervals shrink as the number of words increase.

This is the mathematical architecture of ***Wordfish***. While ***Wordscore*** is running in ***STATA***, ***Wordfish*** is implemented as a computer program written for the ***R*** statistical language.[2] The program requires a word frequencies matrix as input. Such matrix can be obtained with user-friendly programs ***Yoshikoder*** and ***jfreq***.[3] The program then takes the word frequency dataset, generates starting values and runs the algorithm. It outputs the party positions along with word weights and party and word fixed-effects. By doing a parametric bootstrap, it is possible to generate confidence intervals, but that procedure can take several days to complete.

Like ***Wordscore***, ***Wordfish*** assumes that the word meaning remains stable over time. Policy position change is made possible through modification of relative frequencies of words, not their meaning. ***Wordfish*** also assumes the principle dimension extracted from texts captures the political contents of those texts. If researchers want estimates of party positions regarding education, they should run the program on specific

[2] For more information, see The R Project for Statistical Computing website.
[3] Both programs are available at https://people.iq.harvard.edu/~wlowe/Software.html.

education documents. That means identifying such documents. That can be troublesome when working with foreign languages or countries. Finally, ***Wordfish***, as well as ***Wordscore***, gives estimates on a single dimension at the time. When analyzing more than one dimension, it is recommended that researchers first define the dimensions ***ex ante*** and, second, use only documents that contain information relevant to that dimension.

References

Dempster, A. P., N. M. Laird and D. B. Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm." Journal of the Royal Statistical Society, Series B (Methodological) **39**(1): 1-38.

Laver, M., K. Benoit and J. Garry (2003). "Extracting Policy Positions from Political Texts Using Words as Data." American Political Science Review **97**(2): 311-331.

McCallum, A. and K. Nigam (1998). A Comparison of Event Models for Naive Bayes Text Classification. AAAI-98 Workshop on Learning for Text Categorization.

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