Tools for Detecting Geographical and Structural Affinities

Martijn Wieling and John Nerbonne

Department of Computational Linguistics, University of Groningen

Small Tools for Cross-Linguistic Research: 16th June 2009, Utrecht

Overview

- An introduction to the Levenshtein distance
- Using the R*u*G/L04 package to visualize geographical patterns
- New research: Co-clustering varieties and sound correspondences simultaneously

Why do we use the Levenshtein distance?

- Main research interest: statistical methods to investigate language and dialect variation
- We try to determine dialect distances by comparing pronunciations between multiple dialects/languages
- One of the most popular and successful methods to determine pronunciation distance is the Levenshtein distance (Levenshtein, 1964)
- Levenshtein distance: the minimum number of insertions, deletions and substitutions to transform one string into the other

Example of the Levenshtein distance

mɔəlkə	delete ə	1
mɔlkə	subst. ɔ/ɛ	1
mɛlkə	delete ə	1
mɛlk	insert ə	1
mɛlək		
		4

m	ɔ	ə	l		k	ə
m	ɛ		l	ə	k	
	1	1		1		1

Calculating dialect distances

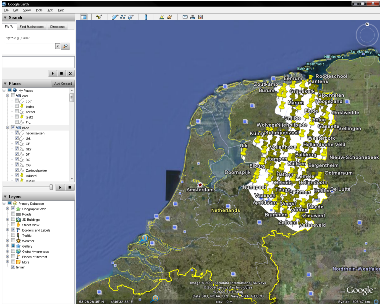
- To determine the aggregate distance between dialects:
  - We determine the distance between each dialect pair for every single word
  - We sum these distances for every word (hundreds of them) and compare them
- Besides dialect distances, this also yields interesting sound correspondences contained in the alignments (more on that later)
  - Note that a 100-word comparison already yields about 500 sound correspondences

Visualizing results

- We can visualize (dialect) distances geographically using the R*u*G/L04 package
  - R*u*G/L04 can be used to visualize any set of distances (lexical, morphological, categorical, ...)
  - The software (including a manual and detailed tutorial) can be freely downloaded from http://www.let.rug.nl/˜kleiweg/L04
  - Note that there is a GUI, but using the commandline is preferred in order to use all options
- In the following we will outline the steps needed to obtain geographical visualizations and show some examples

Generating geographical visualizations (1/3)

- Download the L04 package from the web
- Draw the region and add all varieties using Google Earth



- Convert the Google Earth output online to the R*u*G/L04 format: http://www.let.rug.nl/˜kleiweg/L04/kml

Generating geographical visualizations (2/3)

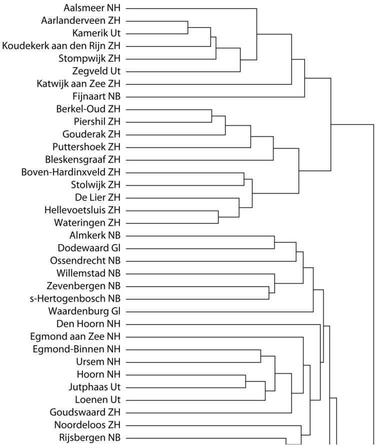
- Obtain the distances between the varieties
  - The Levenshtein algorithm is included in R*u*G/L04
  - But you can also use your own pairwise distances
- To use the Levenshtein algorithm, the following input is needed:
  - One file per word with the pronunciation per location
  - One file which specifies the distance between sounds

Generating geographical visualizations (3/3)

- The following maps can be generated based on the distances:
  - Cluster maps and dendrograms
  - Fuzzy cluster border maps
  - Line maps
  - Vector maps
  - Multidimensional Scaling (MDS) maps
- The following examples are based on Dutch pronunciation data from the Goeman-Taeldeman-Van Reenen-Project data (GTRP; Goeman and Taeldeman, 1996)
  - We use 562 words for 424 varieties in the Netherlands

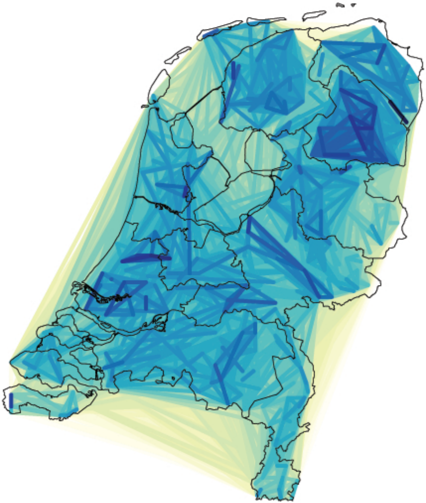
Cluster map

Part of a dendrogram



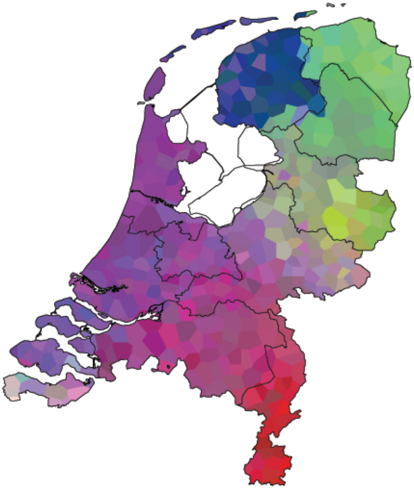
Fuzzy cluster border map

Line map



Vector map

MDS map



Co-clustering varieties and sound correspondences

- Important method in investigating dialectal variation: cluster similar (dialectal) varieties together
- Problem: clustering varieties does not yield a linguistic basis
- Previous solutions: investigate sound correspondences *post hoc* (e.g., Heeringa, 2004)
- New research: Co-clustering to cluster varieties and sound correspondences simultaneously
  - Based on the spectrum of a graph

Obtaining sound correspondences

- Sound correspondences were obtained using the Levenshtein algorithm using a Pointwise Mutual Information procedure (Wieling et al., 2009; included in R*u*G/L04)
  - Levenshtein algorithm:

l	ɛ	ɪ	k		ə	n
l		i	k	h	ɘ	n
	1	1		1	1	

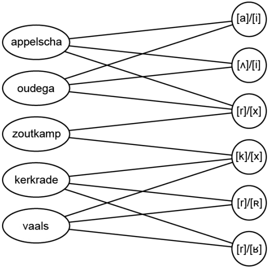
  - Segment distances based on Pointwise Mutual Information:

$$\mathrm{PMI}(x, y) = \log_2\left(\frac{p(x, y)}{p(x)\,p(y)}\right)$$

Generating a bipartite graph from alignments

- A bipartite graph is a graph whose vertices can be divided in two disjoint sets where every edge connects a vertex from one set to a vertex in another set. Vertices within a set are not connected.
- From the alignments, we extract the number of sound correspondences for each variety (compared to a reference site)
- We generated a bipartite graph of varieties v and sound correspondences s
  - There is an edge between v_i and s_j iff freq(s_j in v_i) > 0

Example of a bipartite graph *A*



	[a]/[i]	[ʌ]/[i]	[r]/[x]	[k]/[x]	[r]/[ʀ]	[r]/[ʁ]
Appelscha	1	1	1	0	0	0
Oudega	1	1	1	0	0	0
Zoutkamp	0	0	1	1	0	0
Kerkrade	0	0	0	1	1	1
Appelscha	0	0	0	1	1	1

Co-clustering procedure

- Used by Dhillon (2001) to co-cluster words and documents
- Based on finding the eigenvectors of the adjacency matrix of a bipartite graph and subsequently using the *k*-means algorithm on the eigenvectors to obtain the two-way clustering
  - The mathematical details are not covered in this talk (but see Wieling and Nerbonne, 2009)
- Note that this procedure is not included in R*u*G/L04
  - However, the cluster maps are visualized using R*u*G/L04

Example of co-clustering a biparte graph (1/3)

- Based on the adjacency matrix ***A***:

	[a]/[i]	[ʌ]/[i]	[r]/[x]	[k]/[x]	[r]/[ʀ]	[r]/[ʁ]
Appelscha	1	1	1	0	0	0
Oudega	1	1	1	0	0	0
Zoutkamp	0	0	1	1	0	0
Kerkrade	0	0	0	1	1	1
Appelscha	0	0	0	1	1	1

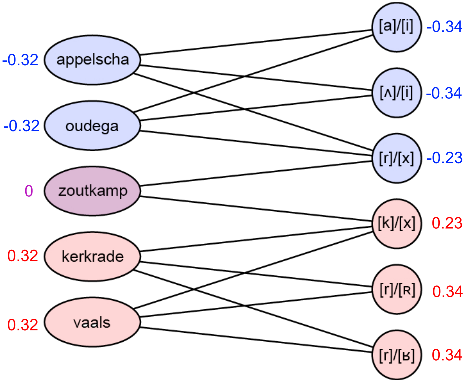
- We can calculate the eigenvectors (of the Laplacian) of ***A***:
  $\lambda_2 = .057$, $\boldsymbol{x} = [-.32\ -.32\ 0\ .32\ .32\ -.34\ -.34\ -.23\ .23\ .34\ .34]^T$
  $\lambda_3 = .53$, $\boldsymbol{x} = [.12\ .12\ -.7\ .12\ .12\ .25\ .25\ -.33\ -.33\ .25\ .25]^T$

Example of co-clustering a biparte graph (2/3)

- To cluster in $k = 2$ groups, we use:
  - λ_2 = .057, $\boldsymbol{x}$ = [-.32 -.32 0 .32 .32 -.34 -.34 -.23 .23 .34 .34]^T

Example of co-clustering a biparte graph (2/3)

- To cluster in $k = 2$ groups, we use:
  - λ_2 = .057, $\boldsymbol{x}$ = [-.32 -.32 0 .32 .32 -.34 -.34 -.23 .23 .34 .34]^T
- We obtain the following co-clustering:

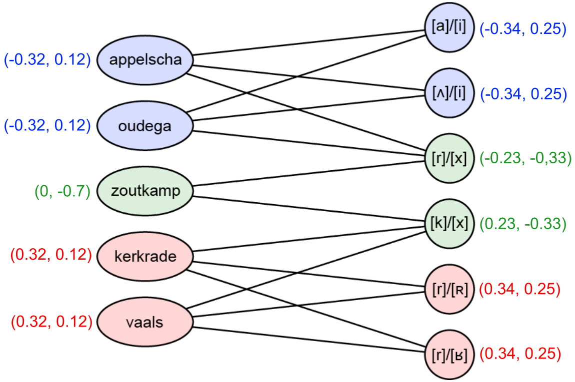


Example of co-clustering a biparte graph (3/3)

- To cluster in $k = 3$ groups, we use:
  - λ_2 = .057, $\boldsymbol{x}$ = [-.32 -.32 0 .32 .32 -.34 -.34 -.23 .23 .34 .34]^T
  - λ_3 = .53, $\boldsymbol{x}$ = [.12 .12 -.7 .12 .12 .25 .25 -.33 -.33 .25 .25]^T

Example of co-clustering a biparte graph (3/3)

- To cluster in $k = 3$ groups, we use:
  - λ_2 = .057, $\boldsymbol{x}$ = [-.32 -.32 0 .32 .32 -.34 -.34 -.23 .23 .34 .34]^T
  - λ_3 = .53, $\boldsymbol{x}$ = [.12 .12 -.7 .12 .12 .25 .25 -.33 -.33 .25 .25]^T
- We obtain the following co-clustering:

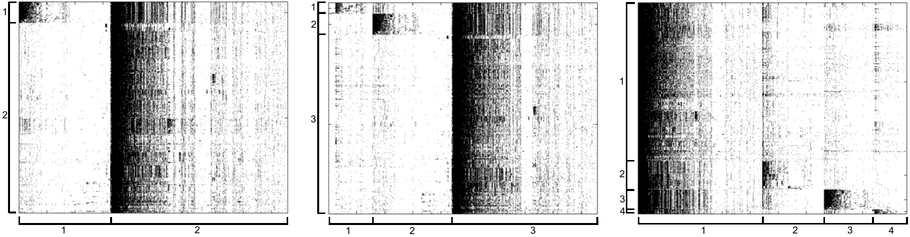


Dataset

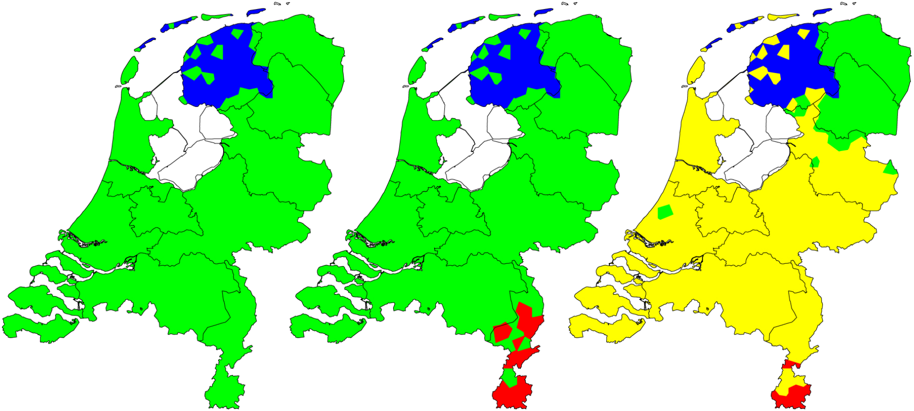
- We used Dutch dialect pronunciations from the GTRP corpus (Goeman and Taeldeman, 1996)
- We generated alignments of pronunciations of 562 words for 424 varieties in the Netherlands against a reference pronunciation
  - The pronunciations of Delft were used as the reference, as we did not have pronunciations of standard Dutch

Distribution of sites

Results: {2,3,4} co-clusters in the data



Results: {2,3,4} clusters of varieties



Results: {2,3,4} clusters of sound correspondences

- Some sound correspondences specific for the Frisian area

Reference	[ʌ]	[ʌ]	[a]	[o]	[u]	[x]	[x]	[r]
Frisian	[ɪ]	[i]	[i]	[ɛ]	[ɛ]	[j]	[z]	[x]

- Some sound correspondences specific for the Limburg area

Reference	[r]	[r]	[k]	[n]	[n]	[w]
Limburg	[ʀ]	[ʁ]	[x]	[ʀ]	[ʁ]	[f]

- Some sound correspondences specific for the Low Saxon area

Reference	[ə]	[ə]	[ə]	[-]	[a]
Low Saxon	[m]	[ŋ]	[ɴ]	[ʔ]	[e]

Discussion

- Bipartite spectral graph partitioning is a very useful method to detect the linguistic basis for the dialectal clustering and can also be used to simultaneously cluster
  - varieties and sound correspondences
  - words and documents
  - genes and conditions
  - ... and ...
- However, there are some limitations:
  - We used transcriptions of a single variety, instead of the standard or proto-language, as the reference pronunciation
  - We did not investigate methods to identify the importance of each sound correspondence
  - Frequency information is discarded

Any questions?

Thank You!

References

- Inderjit Dhillon. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 269–274. ACM New York.
- Ton Goeman, and Johan Taeldeman. 1996. Fonologie en morfologie van de Nederlandse dialecten. Een nieuwe materiaalverzameling en twee nieuwe atlasprojecten. *Taal en Tongval*, 48:38–59.
- Vladimir Levenshtein. 1965. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Adademii Nauk SSSR*, 164:845–848.
- Wilbert Heeringa. 2004. *Measuring Dialect Pronunciation Differences using Levenshtein Distance*. Ph.D. thesis, Rijksuniversiteit Groningen.
- Martijn Wieling and John Nerbonne. 2009. Bipartite spectral graph partitioning to co-cluster varieties and sound correspondences in dialectology. In: Monojit Choudhury (ed.) *Proceedings of the TextGraphs-4 Workshop at the 47th Meeting of the Association for Computational Linguistics*, August 2009, Singapore. Available via http://www.martijnwieling.nl.
- Martijn Wieling, Jelena Prokić, and John Nerbonne. 2009. Evaluating the pairwise string alignment of pronunciations. In: Lars Borin and Piroska Lendvai (eds.) *Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education* (LaTeCH - SHELT&R 2009) Workshop at the 12th Meeting of the European Chapter of the Association for Computational Linguistics. Athens, 30 March 2009, pp. 26-34. Available via http://www.martijnwieling.nl.