Bipartite spectral graph partitioning to co-cluster varieties and sound correspondences

Martijn Wieling

Department of Computational Linguistics, University of Groningen

Seminar in Methodology and Statistics - May 20, 2009

Goal

- Making the title of this presentation understandable!

  Bipartite spectral graph partitioning to co-cluster varieties and sound correspondences

Overview

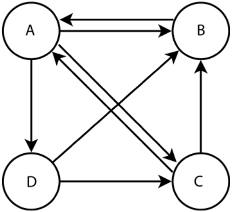
- Why co-clustering?
- Method
  - Introduction to eigenvalues and eigenvectors
  - Simple clustering
  - Co-clustering
- Complete dataset
- Results
- Conclusions

Why co-clustering?

- Research interest: language and dialectal variation
- Important method: 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)
- Co-clustering: clusters varieties and sound correspondences simultaneously
    - Eigenvalues and eigenvectors are central in this approach

Graphs and matrices

- A graph is a set of vertices connected with edges:



- A graph can also be represented by its adjacency matrix $\boldsymbol{A}$

	A	B	C	D
A	0	1	1	1
B	1	0	0	0
C	1	1	0	0
D	0	1	1	0

Eigenvalues and eigenvectors

- The eigenvalues λ and the eigenvectors $\boldsymbol{x}$ of a *square* matrix $\boldsymbol{A}$ are defined as follows:

$$\boldsymbol{Ax} = \lambda \boldsymbol{x} \qquad [\Rightarrow (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \mathbf{0}]$$

- In matrix-form:

$$\begin{bmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- This is solved when:

$$(a_{11} - \lambda)x_1 + a_{12}x_2 = 0$$

$$a_{21}x_1 + (a_{22} - \lambda)x_2 = 0$$

Eigenvalues and eigenvectors

- The eigenvalues λ and the eigenvectors $\boldsymbol{x}$ of a *square* matrix $\boldsymbol{A}$ are defined as follows:

$$\boldsymbol{Ax} = \lambda \boldsymbol{x} \qquad [\Rightarrow (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \mathbf{0}]$$

- In matrix-form:

$$\begin{bmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- This is solved when:

$$(a_{11} - \lambda)x_1 + a_{12}x_2 = 0$$

$$a_{21}x_1 + (a_{22} - \lambda)x_2 = 0$$

Eigenvalues and eigenvectors

- The eigenvalues λ and the eigenvectors $\boldsymbol{x}$ of a *square* matrix $\boldsymbol{A}$ are defined as follows:

$$\boldsymbol{Ax} = \lambda \boldsymbol{x} \qquad [\Rightarrow (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}]$$

- In matrix-form:

$$\begin{bmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- This is solved when:

$$(a_{11} - \lambda)x_1 + a_{12}x_2 = 0$$

$$a_{21}x_1 + (a_{22} - \lambda)x_2 = 0$$

Example of calculating eigenvalues and eigenvectors

- Consider the following example: $\boldsymbol{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$
- Using $(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$ we get:

$$\begin{bmatrix} 1-\lambda & 2 \\ 2 & 1-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Solved when det($\boldsymbol{A}$) = 0: $(1-\lambda)^2 - 4 = 0$
- Using $\lambda_1 = 3$ and $\lambda_2 = -1$ we obtain $\boldsymbol{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\boldsymbol{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Example of calculating eigenvalues and eigenvectors

- Consider the following example: $\boldsymbol{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$
- Using $(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$ we get:

$$\begin{bmatrix} 1-\lambda & 2 \\ 2 & 1-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Solved when det($\boldsymbol{A}$) = 0: $(1-\lambda)^2 - 4 = 0$
- Using $\lambda_1 = 3$ and $\lambda_2 = -1$ we obtain $\boldsymbol{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\boldsymbol{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Example of calculating eigenvalues and eigenvectors

- Consider the following example: $\boldsymbol{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$
- Using $(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$ we get:

$$\begin{bmatrix} 1-\lambda & 2 \\ 2 & 1-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Solved when det($\boldsymbol{A}$) = 0: $(1-\lambda)^2 - 4 = 0$
- Using $\lambda_1 = 3$ and $\lambda_2 = -1$ we obtain $\boldsymbol{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\boldsymbol{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Example of calculating eigenvalues and eigenvectors

- Consider the following example: $\boldsymbol{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$
- Using $(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$ we get:

$$\begin{bmatrix} 1-\lambda & 2 \\ 2 & 1-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Solved when det($\boldsymbol{A}$) = 0: $(1-\lambda)^2 - 4 = 0$
- Using $\lambda_1 = 3$ and $\lambda_2 = -1$ we obtain $\boldsymbol{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\boldsymbol{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Spectrum of a graph

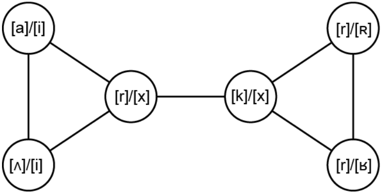
- The spectrum of a graph are the eigenvalues of the adjacency matrix ***A*** of the graph
- The spectrum is considered to capture important structural properties of a graph (Chung, 1997)
- Some interesting applications of eigenvalues and eigenvectors:
  - Principal Component Analysis (PCA; Duda et al., 2001: 114–117)
  - Pagerank (Google; Brin and Page, 1998)
  - Partitioning (i.e. clustering; Von Luxburg, 2007)

Example of spectral graph clustering (1/8)

- Consider the matrix ***A*** with sound correspondences:

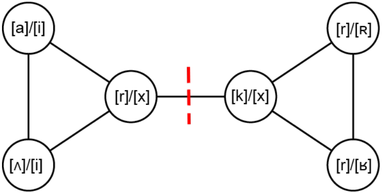
	[a]/[i]	[ʌ]/[i]	[r]/[x]	[k]/[x]	[r]/[ʀ]	[r]/[ʁ]
[a]/[i]	0	1	1	0	0	0
[ʌ]/[i]	1	0	1	0	0	0
[r]/[x]	1	1	0	1	0	0
[k]/[x]	0	0	1	0	1	1
[r]/[ʀ]	0	0	0	1	0	1
[r]/[ʁ]	0	0	0	1	1	0

- In graph-form:



Example of spectral graph clustering (2/8)

- To partition this graph, we have to determine the optimal cut:



- The optimal cut yielding balanced clusters is obtained by finding the eigenvectors of the normalized Laplacian: $\boldsymbol{L_n} = \boldsymbol{D}^{-1}\boldsymbol{L}$, with $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ and $\boldsymbol{D}$ the degree matrix of $\boldsymbol{A}$ (Shi and Malik, 2000; Von Luxburg, 2007).

Example of spectral graph clustering (3/8)

- The adjacency matrix ***A***:

	[a]/[i]	[ʌ]/[i]	[r]/[x]	[k]/[x]	[r]/[ʀ]	[r]/[ʁ]
[a]/[i]	0	1	1	0	0	0
[ʌ]/[i]	1	0	1	0	0	0
[r]/[x]	1	1	0	1	0	0
[k]/[x]	0	0	1	0	1	1
[r]/[ʀ]	0	0	0	1	0	1
[r]/[ʁ]	0	0	0	1	1	0

Example of spectral graph clustering (4/8)

- The Laplacian matrix $\boldsymbol{L}$:

	[a]/[i]	[ʌ]/[i]	[r]/[x]	[k]/[x]	[r]/[ʀ]	[r]/[ʁ]
[a]/[i]	2	-1	-1	0	0	0
[ʌ]/[i]	-1	2	-1	0	0	0
[r]/[x]	-1	-1	3	-1	0	0
[k]/[x]	0	0	-1	3	-1	-1
[r]/[ʀ]	0	0	0	-1	2	-1
[r]/[ʁ]	0	0	0	-1	-1	2

Example of spectral graph clustering (5/8)

- The normalized Laplacian matrix $\boldsymbol{L}_n$:

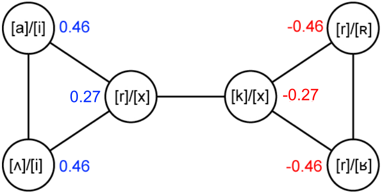
	[a]/[i]	[ʌ]/[i]	[r]/[x]	[k]/[x]	[r]/[ʀ]	[r]/[ʁ]
[a]/[i]	1	-0.5	-0.5	0	0	0
[ʌ]/[i]	-0.5	1	-0.5	0	0	0
[r]/[x]	-0.33	-0.33	1	-0.33	0	0
[k]/[x]	0	0	-0.33	1	-0.33	-0.33
[r]/[ʀ]	0	0	0	-0.5	1	-0.5
[r]/[ʁ]	0	0	0	-0.5	-0.5	1

Example of spectral graph clustering (6/8)

- The eigenvalues λ and eigenvectors $\boldsymbol{x}$ of $\boldsymbol{L}_n$ (i.e. $\boldsymbol{L}_n\boldsymbol{x} = \lambda\boldsymbol{x}$):
  - λ_1 = 0 with $\boldsymbol{x}$ = [-0.41 -0.41 -0.41 -0.41 -0.41 -0.41]^T
  - λ_2 = 0.21 with $\boldsymbol{x}$ = [0.46 0.46 0.27 -0.27 -0.46 -0.46]^T
  - λ_3 = 1.17 with $\boldsymbol{x}$ = [0.36 0.36 -0.49 -0.49 0.36 0.36]^T
  - ...
- The first (smallest) eigenvector does not yield clustering information. Does the second?

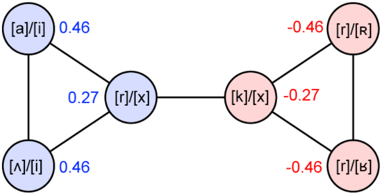
Example of spectral graph clustering (6/8)

- The eigenvalues λ and eigenvectors $\boldsymbol{x}$ of $\boldsymbol{L}_n$ (i.e. $\boldsymbol{L}_n\boldsymbol{x} = \lambda\boldsymbol{x}$):
  - λ_1 = 0 with $\boldsymbol{x}$ = [-0.41 -0.41 -0.41 -0.41 -0.41 -0.41]^T
  - λ_2 = 0.21 with $\boldsymbol{x}$ = [0.46 0.46 0.27 -0.27 -0.46 -0.46]^T
  - λ_3 = 1.17 with $\boldsymbol{x}$ = [0.36 0.36 -0.49 -0.49 0.36 0.36]^T
  - ...
- The first (smallest) eigenvector does not yield clustering information. Does the second? Yes!



Example of spectral graph clustering (7/8)

- If we use the *k*-means algorithm (i.e. minimize the within-cluster sum of squares; Lloyd, 1982) to cluster the eigenvector in two groups we obtain the following partitioning:



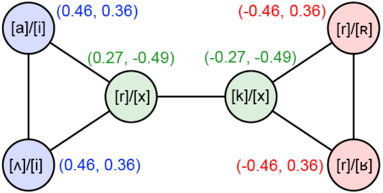
- To cluster in $k > 2$ groups we use the second to k (smallest) eigenvectors

Example of spectral graph clustering (8/8)

- To cluster in $k = 3$ groups, we use:
  - λ_2 = 0.21 with $\boldsymbol{x}$ = $[0.46\ 0.46\ 0.27\ \text{-}0.27\ \text{-}0.46\ \text{-}0.46]^T$
  - λ_3 = 1.17 with $\boldsymbol{x}$ = $[0.36\ 0.36\ \text{-}0.49\ \text{-}0.49\ 0.36\ 0.36]^T$
- We obtain the following clustering:

Example of spectral graph clustering (8/8)

- To cluster in $k = 3$ groups, we use:
  - λ_2 = 0.21 with $\boldsymbol{x}$ = [0.46 0.46 0.27 -0.27 -0.46 -0.46]^T
  - λ_3 = 1.17 with $\boldsymbol{x}$ = [0.36 0.36 -0.49 -0.49 0.36 0.36]^T
- We obtain the following clustering:



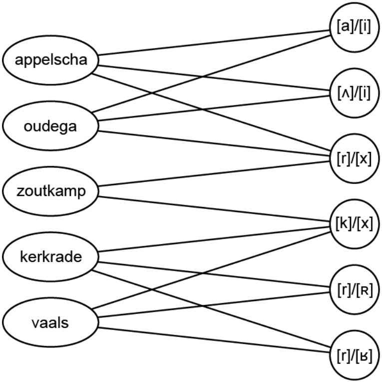
Bipartite graphs

- 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.
- A matrix representation of a bipartite graph:

	[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

Example of co-clustering a biparte graph (1/6)

- The (naive) co-clustering procedure is equal to clustering in one dimension (i.e. cluster eigenvector(s) of normalized Laplacian)
- Consider the following graph:



Example of co-clustering a biparte graph (2/6)

- The adjacency matrix $\boldsymbol{A}$:

	appelscha	oudega	zoutkamp	kerkrade	vaals	[a]/[i]	[ʌ]/[i]	[r]/[x]	[k]/[x]	[r]/[ʀ]	[r]/[ʁ]
appelscha	0	0	0	0	0	1	1	1	0	0	0
oudega	0	0	0	0	0	1	1	1	0	0	0
zoutkamp	0	0	0	0	0	0	0	1	1	0	0
kerkrade	0	0	0	0	0	0	0	0	1	1	1
appelscha	0	0	0	0	0	0	0	0	1	1	1
[a]/[i]	1	1	0	0	0	0	0	0	0	0	0
[ʌ]/[i]	1	1	0	0	0	0	0	0	0	0	0
[r]/[x]	1	1	1	0	0	0	0	0	0	0	0
[k]/[x]	0	0	1	1	1	0	0	0	0	0	0
[r]/[ʀ]	0	0	0	1	1	0	0	0	0	0	0
[r]/[ʁ]	0	0	0	1	1	0	0	0	0	0	0

Example of co-clustering a biparte graph (3/6)

- The Laplacian matrix $\boldsymbol{L}$:

	appelscha	oudega	zoutkamp	kerkrade	vaals	[a]/[i]	[ʌ]/[i]	[r]/[x]	[k]/[x]	[r]/[ʀ]	[r]/[ʁ]
appelscha	3	0	0	0	0	-1	-1	-1	0	0	0
oudega	0	3	0	0	0	-1	-1	-1	0	0	0
zoutkamp	0	0	2	0	0	0	0	-1	-1	0	0
kerkrade	0	0	0	3	0	0	0	0	-1	-1	-1
appelscha	0	0	0	0	3	0	0	0	-1	-1	-1
[a]/[i]	-1	-1	0	0	0	2	0	0	0	0	0
[ʌ]/[i]	-1	-1	0	0	0	0	2	0	0	0	0
[r]/[x]	-1	-1	-1	0	0	0	0	3	0	0	0
[k]/[x]	0	0	-1	-1	-1	0	0	0	3	0	0
[r]/[ʀ]	0	0	0	-1	-1	0	0	0	0	2	0
[r]/[ʁ]	0	0	0	-1	-1	0	0	0	0	0	2

Example of co-clustering a biparte graph (4/6)

- The normalized Laplacian matrix L_n:

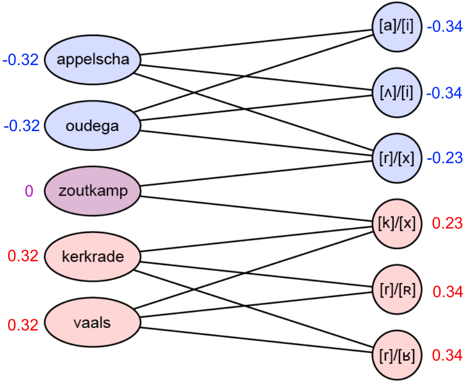
	appelscha	oudega	zoutkamp	kerkrade	vaals	[a]/[i]	[ʌ]/[i]	[r]/[x]	[k]/[x]	[r]/[ʀ]	[r]/[ʁ]
appelscha	1	0	0	0	0	-0.33	-0.33	-0.33	0	0	0
oudega	0	1	0	0	0	-0.33	-0.33	-0.33	0	0	0
zoutkamp	0	0	1	0	0	0	0	-0.5	-0.5	0	0
kerkrade	0	0	0	1	0	0	0	0	-0.33	-0.33	-0.33
appelscha	0	0	0	0	1	0	0	0	-0.33	-0.33	-0.33
[a]/[i]	-0.5	-0.5	0	0	0	1	0	0	0	0	0
[ʌ]/[i]	-0.5	-0.5	0	0	0	0	1	0	0	0	0
[r]/[x]	-0.33	-0.33	-0.33	0	0	0	0	1	0	0	0
[k]/[x]	0	0	-0.33	-0.33	-0.33	0	0	0	1	0	0
[r]/[ʀ]	0	0	0	-0.5	-0.5	0	0	0	0	1	0
[r]/[ʁ]	0	0	0	-0.5	-0.5	0	0	0	0	0	1

Example of co-clustering a biparte graph (5/6)

- 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 (5/6)

- 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:

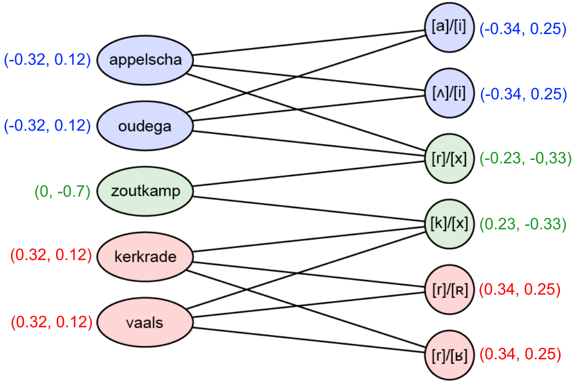


Example of co-clustering a biparte graph (6/6)

- 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 (6/6)

- 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:



A faster method

- The previous method is relatively slow due to the use of the large (sparse) matrix $\boldsymbol{A}$ of size $(n+m) \times (n+m)$
- The matrix $\boldsymbol{A}'$ contains the same information, but is more dense (size: $n \times m$):

	[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

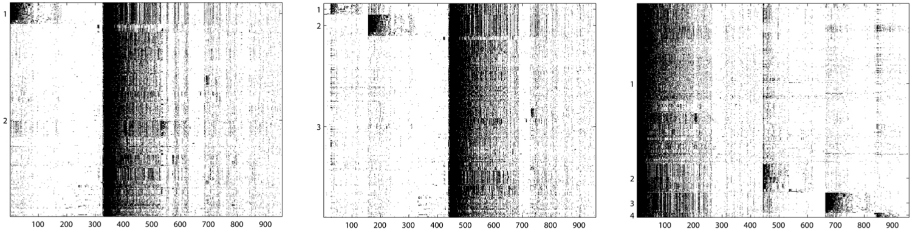
- The singular value decomposition (SVD) of $\boldsymbol{A}'_n$ also results in equivalent clustering information and is quicker to compute (Dhillon, 2001)

Complete dataset

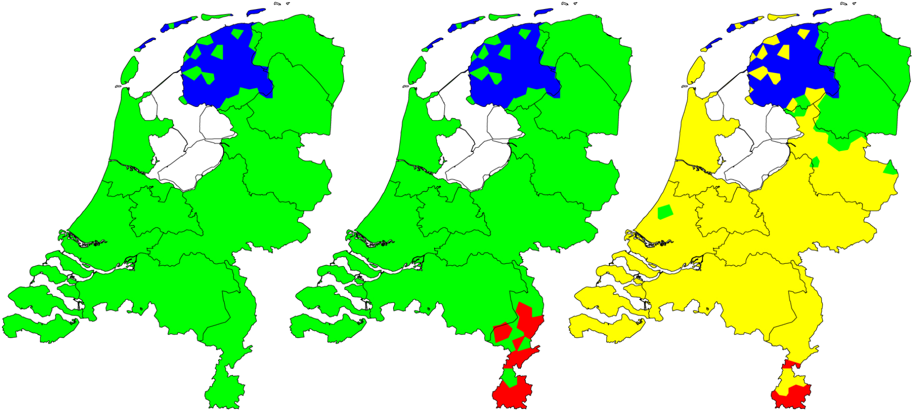
- Alignments of pronunciations of 562 words for 424 varieties in the Netherlands against a reference pronunciation
  - Pronunciations originate from the GTRP (Goeman and Taeldeman, 1996; Van den Berg, 2003; Wieling et al., 2007)
  - The pronunciations of Delft were used as the reference
  - Alignments were obtained using the Levenshtein algorithm using a Pointwise Mutual Information procedure (Wieling et al., 2009)
- 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
- All the following co-clustering results are obtained applying the fast SVD method

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

- 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]

- Sound correspondences specific for the Limburg area

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

- Sound correspondences specific for the Low Saxon area

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

To conclude

- Bipartite spectral graph partitioning is a very useful method to simultaneously cluster
  - varieties and sound correspondences
  - words and documents
  - genes and conditions
  - ... and ...
- Do you now understand the title?
  - Bipartite Spectral graph partitioning to co-cluster varieties and sound correspondences

To conclude

- Bipartite spectral graph partitioning is a very useful method to simultaneously cluster
  - varieties and sound correspondences
  - words and documents
  - genes and conditions
  - ... and ...
- Do you now understand the title?
  - Bipartite Spectral graph partitioning to co-cluster varieties and sound correspondences

Any questions?

Thank You!

References

- Boudewijn van den Berg. 2003. *Phonology & Morphology of Dutch & Frisian Dialects in 1.1 million transcriptions*. Goeman-Taeldeman-Van Reenen project 1980–1995, Meertens Instituut Electronic Publications in Linguistics 3. Meertens Instituut (CD-ROM), Amsterdam.
- Sergey Brin, and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computers Networks and ISDN Systems*, 30(1–7):107–117.
- Fan Chung. 1997. *Spectral Graph Theory*. American Mathematical Society.
- 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.
- Richard Duda, Peter Hart, and David Stork. 2001. *Pattern Classification*. Wiley 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.
- Wilbert Heeringa. 2004. *Measuring Dialect Pronunciation Differences using Levenshtein Distance*. Ph.D. thesis, Rijksuniversiteit Groningen.
- Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.
- Ulrike von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):495–416.
- Jianbo Shi, and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888-905.
- Martijn Wieling, Wilbert Heeringa, and John Nerbonne. 2007. An aggregate analysis of pronunciation in the Goeman-Taeldeman-Van Reenen-Project data. *Taal en Tongval*, 59(1):84–116.
- 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.