Matrix Computations for Pagerank and Related Semantic Graphs

Lars Eldén

Linköping

ETH October 2008

Overview

Semantic graph

Semantic graph

"A semantic graph organizes relational data by using nodes to represent entities and edges to connect related entities. Hidden relationships in the data are then uncovered by examining the structure and properties of the semantic graph."

T. Kolda et al, Data Sciences Technology for Homeland Security Information Management and Knowledge Discovery, Sandia, [16]

General definition, perhaps too general?

Google search 2005: university 2 billion hits

Top hits:

`Harvard, Stanford, Cambridge, Yale, Cornell, Oxford`

2008 search results are more difficult to understand

Google Mathematics

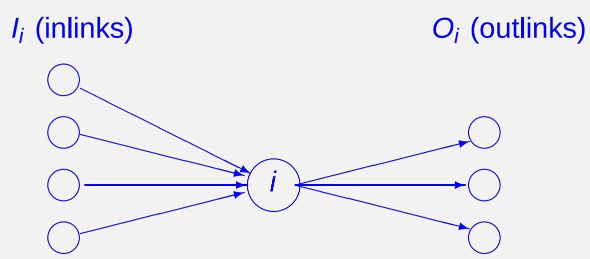
- Google ranks 10^{10} web pages ??
- The ranking is partly based on a mathematical model of the link structure of the Internet
- The ranking is recomputed each month (?) and the computation takes 1-2 weeks(?): the world's largest matrix computation
- When Google started 1998 there was some mathematical theory but not enough. It was found experimentally(?) that the method works.
- Now there is quite some literature on pagerank and its computation.

Search Engines

Web crawler: Computer program that downloads web pages and scans their contents
Search engine:

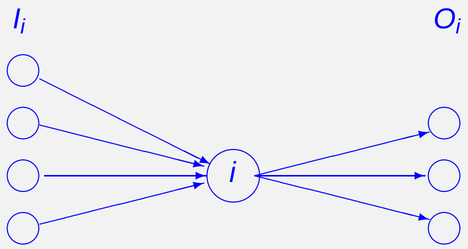
1. Scan the page and collect key words (indexing)
2. Find the link structure graph of the whole Internet and store it as a sparse matrix (dimension $n \approx 10^{10}$?)
3. Rank all web pages

Links



All web pages are ordered: $1, 2, \ldots, n$
Pagerank of page i: a number r_i between 0 and 1 that indicates how important the page is.

Links, cont.

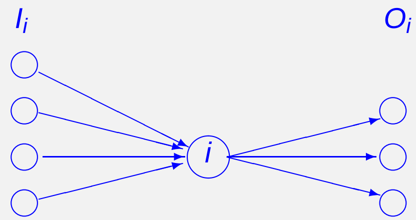


Provisional definition of pagerank: The more **inlinks** your page has, the more important it is.
Easy to manipulate: create pages that point to yours

New provisional definition (Brin and Page 1998 [4])

The more inlinks from important pages your page has, the more important it is

$$r_i = \sum_{j \in I_i} \frac{r_j}{N_j}, \qquad i = 1, 2, \ldots, n.$$



In the sum: The rank of page j is divided equally between is outlinks.

Questions

Recursive definition:

$$r_i = \sum_{j \in I_i} \frac{r_j}{N_j}, \qquad i = 1, 2, \ldots, n.$$

1. Does there exist a solution such that $0 \leq r_i \leq 1$?
2. If it exists, how to compute it?

Suggested Algorithm: iterate

1. Guess starting values $r_i^{(0)}$, $i = 1, \ldots, n$.
2. Iterate until convergence:

$$r_i^{(k)} = \sum_{j \in I_i} \frac{r_j^{(k-1)}}{N_j}, \qquad i = 1, 2, \ldots, n.$$

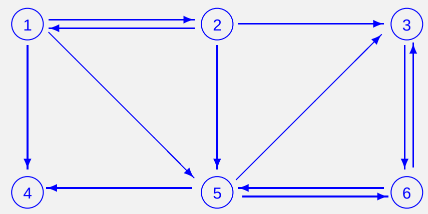
3. If the change is small enough, then stop. Otherwise go back to 2.

Matrix Formulation

$$Q_{ij} = \begin{cases} 1/N_j & \text{if there is a link from } j \text{ to } i \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{array}{cc} & j \\ i & \begin{pmatrix} & & & * & & \\ & & & 0 & & \\ & & & \vdots & & \\ 0 & * & \cdots & * & * & \cdots \\ & & & \vdots & & \\ & & & 0 & & \\ & & & * & & \end{pmatrix} \leftarrow \text{inlinks} \\ & \uparrow \\ & \text{outlinks} \end{array}$$

Example



$$Q = \begin{pmatrix} 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{3} & 0 \end{pmatrix}$$

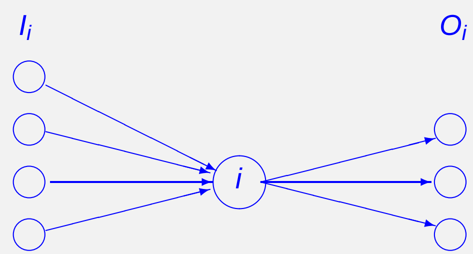
No outlinks from page 4: corresponding column is equal to zero.
All other columns: the elements sum up to 1

Random Walk Model

- **Random surfer:** On each page, choose one of the outlinks with equal probability
- Observe the surfer for a long time and count the frequency of visits to all pages
- **Pagerank r_i:** the asymptotic probability that the surfer is at page i.
- **Markov chain:** Discrete time and no memory. The state at time $t+1$ depends on the state at time t only.
- Q is the matrix of transition probabilities

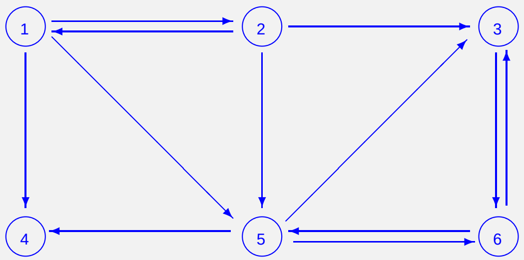
Pagerank definition:

$$r_i = \sum_{j \in I_i} \frac{r_j}{N_j}, \qquad i = 1, 2, \ldots, n.$$



r_i: relative frequency of visits at page i

Example, cont.



The model does not work: The surfer gets stuck at page 4.
Remedy: Introduce (artificially) links from 4 to all the others.
The random walk can continue forever.

Modified Transition Matrix

Define

$$d_j = \begin{cases} 1 & \text{if } N_j = 0 \\ 0 & \text{otherwise,} \end{cases}$$

for $j = 1, \ldots, n$, and

$$e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^n.$$

$$P = Q + \frac{1}{n} e d^T.$$

P is a proper **column-stochastic matrix**: Non-negative elements, and the elements of each column sum up to 1.

Column-Stochastic Matrix

Proposition

A column-stochastic matrix P satisfies

$$e^T P = e^T, \qquad e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Equivalently: e is an eigenvector of P^T with the eigenvalue 1.

Eigenvector of P

Recursive definition of pagerank:

$$r_i = \sum_{j \in I_i} \frac{r_j}{N_j}, \qquad i = 1, 2, \ldots, n.$$

Now equivalent to

$$Pr = r.$$

We want to **find this eigenvector of** P (not P^T)
Questions:

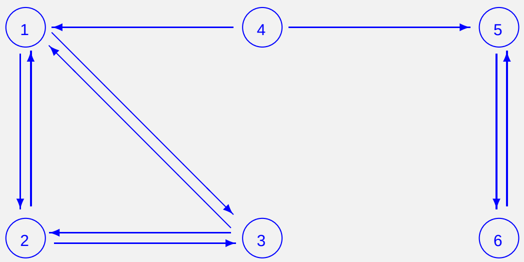
1. Is the eigenvector r unique?
2. Is it a probability vector?

Perron-Frobenius Theory

Theorem

If P is an **irreducible** *column-stochastic matrix, then the largest eigenvalue is equal to 1, and the corresponding eigenvector r has non-negative elements.*

The Google matrix is **reducible**: the random walk can get stuck in subgraphs of the Internet



Corresponding matrix

$$P = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Eigenvalues: $1, 1, -1, -0.5, -0.5$

Irreducible Matrix

$$A = \alpha P + (1-\alpha)\frac{1}{n}ee^T, \qquad 0 \le \alpha \le 1$$

A is column-stochastic:

$$e^T A = \alpha e^T P + (1-\alpha)\frac{1}{n}e^T ee^T = \alpha e^T + (1-\alpha)e^T = e^T.$$

Random walk interpretation

The surfer will jump to a random page with probability $1-\alpha$

Teleportation

Theorem

The positive column-stochastic matrix $A = \alpha P + (1-\alpha)\frac{1}{n}ee^T$ has a unique eigenvector $r > 0$ with the eigenvalue 1.

Power Method

The power method for $Ar = \lambda r$

for $k = 1, 2, \ldots$

$q^{(k)} = Ar^{(k-1)}$

$r^{(k)} = q^{(k)}/\|q^{(k)}\|_1$

Expand initial approximation $r^{(0)}$ in terms of eigenvectors:

$$r^{(0)} = c_1 r_1 + c_2 r_2 + \cdots + c_n r_n,$$

$$A^k r^{(0)} = \lambda_1^k \left(c_1 r_1 + \sum_{j=2}^{n} c_j \left(\frac{\lambda_j}{\lambda_1}\right)^k r_j \right).$$

Second Eigenvalue of A

Rate of convergence in power method

$$\left|\frac{\lambda_2}{\lambda_1}\right|^k$$

Theorem

Given that the eigenvalues of the column-stochastic matrix P are $\{1, \lambda_2, \lambda_3 \ldots, \lambda_n\}$, the eigenvalues of $A = \alpha P + (1-\alpha)\frac{1}{n}ee^T$ are $\{1, \alpha\lambda_2, \alpha\lambda_3, \ldots, \alpha\lambda_n\}$.

Standard choice: $\alpha = 0.85$

Computing $y = Az$

$$A = \alpha P + (1-\alpha)\frac{1}{n}ee^T \in \mathbb{R}^{n\times n}$$

Recall: $n \approx 10^{10}$

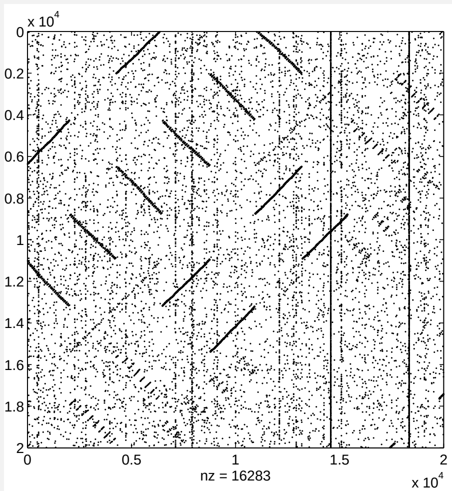
- Cannot form A (dense)
- Cannot form $P = Q + \frac{1}{n}ed^T$ (many dense columns) (Q is the actual link matrix)

Matlab code:

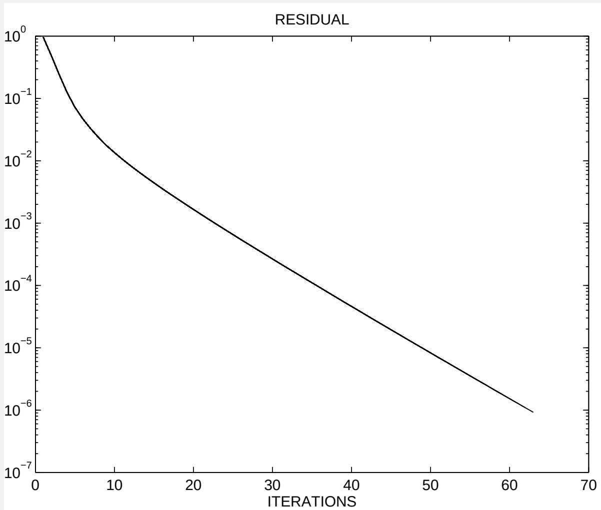
```
y=alpha*Q*z;
beta=1-norm(y,1);
y=y+beta*v;
residual=norm(y-z,1);
```

Example: Link structure for `stanford.edu`

Web pages: 281903, links: 2312497.
Sparsity: 0.0029%

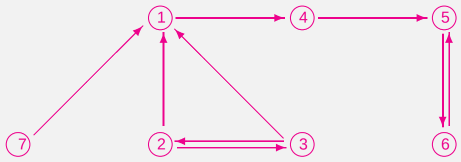


Convergence with $\alpha = 0.85$



What happens when $\alpha \to 1$?

$\alpha \to 0$: completely random, all pages get the same rank
$\alpha \to 1$: The links determine the rank
BUT: counterintuitive effects



Node	0.5	0.85	0.9999
1	0.15	0.07	0.0
2	0.10	0.04	0.0
3	0.10	0.04	0.0
4	0.15	0.08	0.0
5	0.24	0.39	0.5
6	0.19	0.36	0.5
7	0.07	0.02	0.0

Is pagerank important?

Perhaps the Google problem as such is not so important. There are other applications where similar methods can be applied: graph similarity, social networks, semantic graphs, etc.

Basic pagerank idea

The more inlinks from important pages your page has, the more important it is:

$$r_i = \sum_{j \in I_i} \frac{r_j}{N_j}, \qquad i = 1, 2, \dots, n.$$

Equivalent to matrix eigenvalue problem

$$Ar = \lambda r$$

Applications of pagerank

- Ranking of doctoral programs [21]
  Recruiting a doctor from a good program is worth more than recruiting from a not so good program
- Ranking of football teams [10, 12]
  Winning over a strong team is worth more than winning over a weaker team
- Bibliometry: Finding scientific gems with Google's PageRank algorithm [5]
  Getting a citation from an important paper is better than getting one from a less important paper
  Counteracts "citation spamming"
- ....

HITS: Hypertext Induced Topic Search

Kleinberg 1999 [13]

- Hub A web page with many outlinks
- Authority A web page with many inlinks
- Good hubs point to good authorities
- Good authorities are pointed to by good hubs
- Hub score u and authority score v

$$u = Av, \qquad v = A^T u$$

A is a graph matrix (adjacency matrix, essentially)

- SVD: $A = U \Sigma V^T$

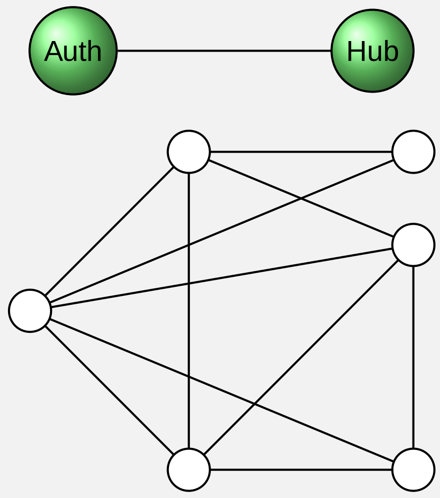
$$u = u_1, \qquad v = v_1$$

First singular vectors

Applications of HITS

- Political science: Supreme Court Precendence [8, 9]
- Summarization [24]
- Synonym extraction [3]
- Generank [20]

Two graphs



Eigenvalue problem

Adjacency matrices of directed hub-authority graphs:

$$E = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \qquad E^T = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

Eigenvalue problem for HITS:

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \sigma \begin{pmatrix} u \\ v \end{pmatrix} \quad \Leftrightarrow \quad \left((E \otimes A) + (E \otimes A)^T \right) \begin{pmatrix} u \\ v \end{pmatrix} = \sigma \begin{pmatrix} u \\ v \end{pmatrix}$$

Interpretation

Eigenvalue problem for product graph.
Element u_j gives a measure of how similar node j in the graph A is to the hub node.

Generalization

Blondel et al. *A Measure of Similarity between Graph Vertices*, SIAM Review, 2004
Adjacency matrices A and B of two directed graphs:
Eigenvalue problem:

$$\left((B \otimes A) + (B \otimes A)^T \right) x = \lambda x$$

Interpretation

Eigenvalue problem for product graph:
The elements of the prinicipal eigenvector gives measures of the similarities of nodes in one graph to the nodes in the other graph

Special case: Self-similarity

$$\left((A \otimes A) + (A \otimes A)^T \right) x = \lambda x$$

Parallel Texts

- The same text in two (or more) languages
- The sentences in both languages are aligned: corresponding sentences are identified.
- Compute a translation of the words in one language to the other.

Purpose: Acquisition of a bilingual lexicon

Text parser

Use a text parser to produce a term-sentence matrix
Publicly available text parsers:

- TMG: Matlab toolbox (Zeimpekis & Gallopoulos)
- GTP: Java or C++ (Giles, Wo & Berry)

Text parsing: vector space model

Sentence 1:	The Google matrix P is a model of the Internet.
Sentence 2:	P_{ij} is nonzero if there is a link from web page j to i.
Sentence 3:	The Google matrix is used to rank all web pages
Sentence 4:	The ranking is done by solving a matrix eigenvalue problem.
Sentence 5:	England dropped out of the top 10 in the **FIFA** ranking.

Term	S1	S2	S3	S4	S5
eigenvalue	0	0	0	1	0
England	0	0	0	0	1
FIFA	0	0	0	0	1
Google	1	0	1	0	0
Internet	1	0	0	0	0
link	0	1	0	0	0
matrix	1	0	1	1	0
page	0	1	1	0	0
rank	0	0	1	1	1
web	0	1	1	0	0

Parallel Texts: Term-sentence matrices

Denote the term-sentence matrices A_1 and A_2
Columns: sentences, rows: terms
The sentences are aligned $\Longrightarrow A_1 \in \mathbb{R}^{m_1 \times n}, \qquad A_2 \in \mathbb{R}^{m_2 \times n}$

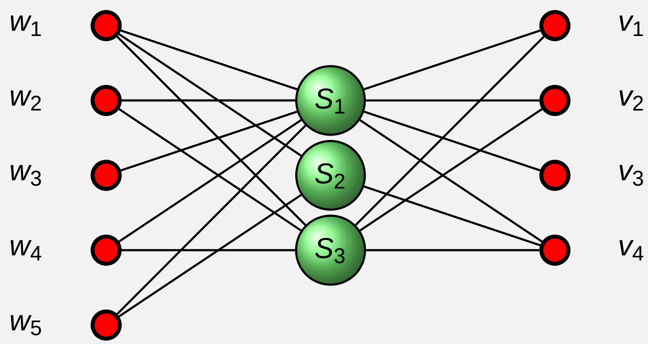
	Lang. 1	Lang. 2	Sentences
Lang. 1	0	0	A_1
Lang. 2	0	0	A_2
Sentences	A_1^T	A_2^T	0

Define

$$A = \begin{pmatrix} 0 & 0 & A_1 \\ 0 & 0 & A_2 \\ A_1^T & A_2^T & 0 \end{pmatrix}$$

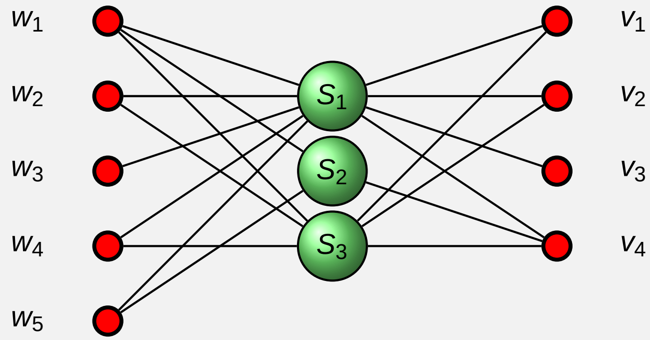
Adjacency matrix of Bipartite graph

$$A = \begin{pmatrix} 0 & 0 & A_1 \\ 0 & 0 & A_2 \\ A_1^T & A_2^T & 0 \end{pmatrix}$$



Hypothesis

The structure of the graph makes it possible to identify which v_i corresponds to w_j $\Longrightarrow$ Dictionary



Graph similarity

Blondel et al. *A Measure of Similarity between Graph Vertices*, SIAM Review, 2004

G undirected graph

Symmetric adjacency matrix $A \in \mathbb{R}^{n \times n}$.

Define the linear operator

$$\mathcal{A} : \mathbb{R}^{n \times n} \ni X \mapsto AXA \in \mathbb{R}^{n \times n}$$

Self-similarity matrix of G:

the dominant eigenvector S_1 of $\mathcal{A}$. ("dominant eigenspace")

Perron-Frobenius theory $\Longrightarrow$ $S_1 \geq 0$

Structured similarity

Type constraints (Fraikin, Van Dooren, 2006)

$$A = \begin{pmatrix} 0 & 0 & A_1 \\ 0 & 0 & A_2 \\ A_1^T & A_2^T & 0 \end{pmatrix}$$

No similarity between sentences and terms:

$$S = \begin{pmatrix} S_{11} & S_{12} & 0 \\ S_{21} & S_{22} & 0 \\ 0 & 0 & S_{33} \end{pmatrix}$$

Hypothesis:

The similarity matrix S_{12} will give the dictionary

Eigenvalue problem with a lot of structure

Let

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} = U \begin{pmatrix} \Sigma & 0 \end{pmatrix} V^T = \begin{pmatrix} U_1 \\ U_2 \end{pmatrix} \Sigma V_1^T$$

Assume $\sigma_1 > \sigma_2 \geq \cdots$

Lemma (Eigenvalues for the directed graph)

The largest eigenvalue of the linear operator corresponding to the directed graph is simple and the corresponding eigenvector is

$$\begin{pmatrix} u_1 u_1^T & 0 \\ 0 & v_1 v_1^T \end{pmatrix}$$

We want to have

$$S = \begin{pmatrix} S_{11} & S_{12} & 0 \\ S_{21} & S_{22} & 0 \\ 0 & 0 & S_{33} \end{pmatrix}$$

Similarity matrix S_{12}

We have the partitioning

$$U = \begin{pmatrix} U_1 \\ U_2 \end{pmatrix}, \qquad u_1 = \begin{pmatrix} u_{11} \\ u_{21} \end{pmatrix}$$

We get

$$S_{12} = u_{11} u_{21}^T$$

Similarity matrix S_{12}: interpretation

$$S_{12} = u_{11} u_{21}^T$$

Rank one approximation of the combined term-sentence matrix:

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \approx \sigma_1 \begin{pmatrix} u_{11} \\ u_{21} \end{pmatrix} v_1^T$$

Compare the size of the elements in the upper and lower part of the first singular vector.

Pair the two largest, the next two, and so on.

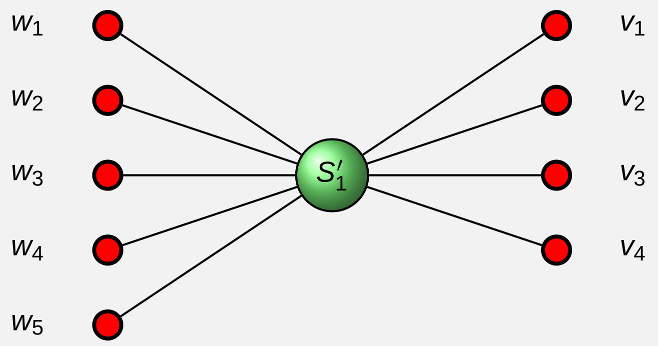
Not likely to be very accurate!

S_{12}: graph interpretation

Rank one approximation of the combined term-sentence matrix:

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \approx \sigma_1 \begin{pmatrix} u_{11} \\ u_{21} \end{pmatrix} v_1^T$$

The graph has been collapsed to one sentence node:



Similarity

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \approx \sigma_1 \begin{pmatrix} u_{11} \\ u_{21} \end{pmatrix} v_1^T, \qquad S_{12} = u_{11} u_{21}^T$$

- SVD constructs a pseudo-sentence with most important contents in the collection
- Similarity based on how much the different words occur in the most important sentence

Not likely to be very accurate for creating the dictionary:

- Reduction to only one sentence is too drastic
- Several topics are treated

Our hypothesis:

Similarity matrix S_{12} gives the dictionary. Not good enough!

Alternative approaches

1. Rank$-k$ approximation with SVD

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \approx \begin{pmatrix} U_1 \\ U_2 \end{pmatrix} \Sigma_k V_k^T, \qquad S_{12} = U_1 \Sigma_k^2 U_2^T$$

   1.1 Advantage: Algorithms for large sparse problems well developed
   1.2 Disadvantage: S_{12} has negative elements, then what is similarity?

2. Nonnegative rank$-k$ approximation

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \approx \begin{pmatrix} W_1 \\ W_2 \end{pmatrix} Z^T, \qquad W_1 \geq 0, W_2 \geq 0, Z \geq 0,$$

   2.1 Advantage: $S_{12} \geq 0$?
   2.2 Disadvantage: Not obvious how to choose S_{12}

Rank$-k$ SVD approximation: interpretation

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} V_k \approx \begin{pmatrix} U_1 \\ U_2 \end{pmatrix} \Sigma_k \qquad S_{12} = U_1 \Sigma_k^2 U_2^T$$

- Rank$-k$ approximation of $\begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$: Choose the k most important topics (orthogonal pseudo-sentences)
- Noise is removed
- For term i in language 1 choose the largest positive number in row i of S_{12} for translation

Related to Fraikin et al, Optimizing the coupling...

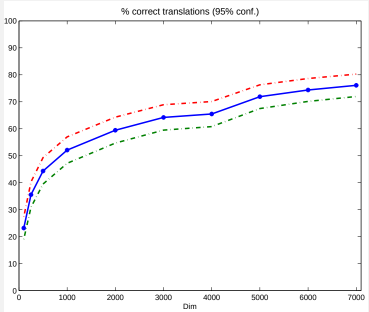
$$\max_{U^TU=I} \operatorname{tr}(U^T A U U^T A^T U) = \max_{U^TU=I} \|U^T A U\|_F^2$$

Numerical test: Europarl corpus

- `http://www.statmt.org/europarl/`
- English & Swedish
- Proceedings of the EU parliament
- $\approx 800,000$ sentences
- Stop words removed, stemming performed
- Only words that occur more than 100 times in the corpus are used $\Longrightarrow$ $\approx$ 6000 words in each language
- Text parser: GTP
- Evaluation of results based on sampling
- Linguistics literature: $\approx 60\%$ accuracy

SVD compression

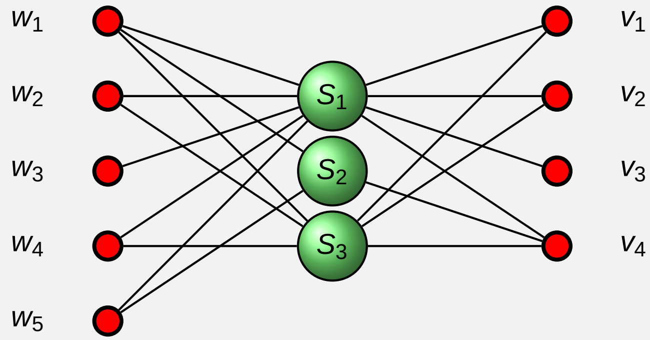
$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \approx U_k \Sigma_k V_k^T$$



BUT Without SVD compression: Almost 90% correct!

Adjacency matrix of Bipartite graph

$$A = \begin{pmatrix} 0 & 0 & A_1 \\ 0 & 0 & A_2 \\ A_1^T & A_2^T & 0 \end{pmatrix}$$



Parallel text $\longrightarrow$ dictionary

Conclusion

The structure of the graph gives enough information to construct a reliable dictionary, but the eigenvector approach does not give enough information. Best results from simple comparison of word vectors:

$$S_{12} = A_1 A_2^T$$

Conclusions

- Graph $\Leftrightarrow$ Sparse matrix
  Powerful techniques from linear algebra: eigenvalues, matrix factorizations (including non-negative)
- Very large problems can be handled
- Recent development: graph with text in each node $\Rightarrow$ tensor problem [14]
- Graphs with multiple linkages $\Rightarrow$ tensor problem [1, 6, 17, 15]

References I

- Brett W. Bader, Richard A. Harshman, and Tamara G. Kolda.
  Temporal analysis of semantic graphs using ASALSAN.
  In *ICDM 2007: Proceedings of the 7th IEEE International Conference on Data Mining*, pages 33–42, October 2007.
- P. Berkin.
  A survey on PageRank computing.
  Internet Mathematics, 2:73–120, 2005.
- V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. Van Dooren.
  A measure of similarity between graph vertices: Applications to synonym extraction and web searching.
  SIAM Review, 46:647–666, 2004.
- S. Brin and L. Page.
  The anatomy of a large-scale hypertextual web search engine.
  Computer Networks and ISDN Systems, 30:107–117, 1998.

References II

- P. Chen, H. Xie, S. Maslov, and S. Redner.
  Finding scientific gems with Google's PageRank algorithm.
  J. Informetrics, 1:8–15, 2007.
- Daniel M. Dunlavy, Tamara G. Kolda, and W. Philip Kegelmeyer.
  Multilinear algebra for analyzing data with multiple linkages.
  Technical Report SAND2006-2079, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, April 2006.
- Lars Eldén.
  A note on the eigenvalues of the Google matrix.
  Technical Report LiTH-MAT-R–04-01, Department of Mathematics, Linköping University, 2004.
- J. H. Fowler and S. Jeon.
  The authority of supreme court precedent.
  Social Networks, to appear, 2007.

References III

- J. H. Fowler, T. R. Johnson, J. F. Spriggs II, S. Jeon, and P. J. Wahlbeck.
  Network analysis and the law: Measuring the legal importance of supreme court precedents.
  Political Analysis, 15:324–346, 2007.
- Anjela Govan, Carl Meyer, and Russell Albright.
  Generalizing Google's PageRank to rank national football league teams.
- R. Horn and S. Serra-Capizzano.
  A general setting for the parametric Google matrix.
  Internet Math., to appear, 2007.
- James P. Keener.
  The Perron–Frobenius theorem and the ranking of football teams.
  SIAM Review, 35(1):80–93, 1993.

References IV

- J. M. Kleinberg.
  Authoritative sources in a hyperlinked environment.
  J. Assoc. Comput. Mach., 46(5):604–632, 1999.
- T. Kolda and B. Bader.
  The TOPHITS model for higher-order web link analysis.
  In *Workshop on Link Analysis, Counterterrorism and Security*, 2006.
- T. Kolda, B. Bader, and J. Kenny.
  Higher-order web link analysis using multilinear algebra.
  In *Proc. 5th IEEE International Conference on Data Mining, ICDM05*, pages 27–30. IEEE Computer Society Press, 2005.

References V

- T. Kolda, D. Brown, J. Corones, T. Critchlow, T. Eliassi-Rad, L. Getoor, B. Hendrickson, V. Kumar, D. Lambert, C. Matarazzo, K. McCurley, M. Merrill, N. Samatova, D. Speck, R. Srikant, J. Thomas, M. Wertheimer, and P. C. Wong.
  Data sciences technology for homeland security information management and knowledge discovery.
  Technical Report SAND2004-6648, Sandia National Laboratories, Jan 2005.
- T. G. Kolda and B. W. Bader.
  Tensor decompositions and applications.
  SIAM Review, 50:??, 2008, to appear.
- A. N. Langville and C. D. Meyer.
  Google's PageRank and Beyond: The Science of Search Engine Rankings.
  Princeton University Press, 2006.

References VI

- C. D. Meyer.
  Matrix Analysis and Applied Linear Algebra.
  SIAM, Philadelphia, 2000.
- J. L. Morrison, R. Breitling, D. J. Higham, and D. R. Gilbert.
  Generank: Using search engine technology for the analysis of microarray experiment.
  BMC Bioinformatics, 6:233, 2005.
- Benjamin M. Schmidt and Matthew M. Chingos.
  Ranking doctoral programs by placement: A new method.
  PS: Political Science & Politics, 40:523–529, 2007.
- T. Törnfeldt.
  Graph similarity, parallel texts, and automated bilingual lexicon acquisition.
  Master's thesis, Linköping Institute of Technology, Department of Mathematics. LiTH-MAT-Ex-08/08-SEbri, 2008.

References VII

- M. Totty and M. Mangalindan.
  As Google becomes Web's gatekeeper, sites fight to get in.
  Wall Street Journal, CCXLI(39)(Feb. 26), 2003.
- H. Zha.
  Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering.
  In *Proc. 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–120, Tampere, Finland, 2002.