Artificial Tutoring with Latent Semantic Analysis

ProLearn Summer School 2006, Bled, Slowenia

Fridolin Wild
Vienna University of Economics and Business Administration

Artificial Tutoring?

- Feedback to Support Competence Development
- Various Automatic Assessment Methods
- Latent Semantic Analysis
- The LSA Package for R
- Demo: Essay Scoring with LSA

Input (e.g., documents)

c1: *Human* machine *interface* for ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user* perceived *response time* to error measurement

m1: The generation of random, binary, ordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

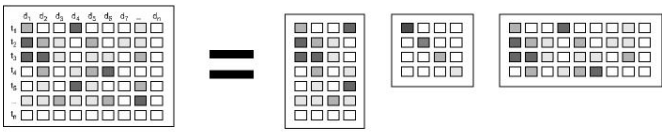
Only the red terms appear in more than one document, so strip the rest.

{ M } =

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Deerwester, Dumais, Furnas, Landauer, and Harshman (1990): Indexing by Latent Semantic Analysis, In: Journal of the American Society for Information Science, 41(6):391-407

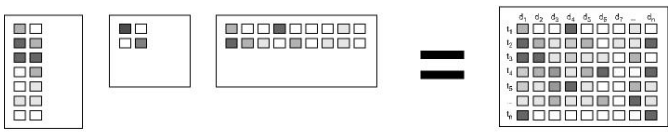
Singular Value Decomposition



$$M \equiv T \; S \; D^T$$

Latent Semantics

- Assumption: language utterances have a semantic structure
- However, this structure is obscured by word usage (noise, synonymy, polysemy, …)
- Proposed LSA Solution: map doc-term matrix using conceptual indices derived statistically (truncated SVD)



$$T_k \, S_k \, D_k^T = M_k$$

Reconstructed, Reduced Matrix

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

m4: *Graph minors*: A *survey*

doc2doc - similarities

unreduced

- Based on $M = TSD'$
- Pearson Correlation over document vectors

	c1	c2	c3	c4	c5	m1	m2	m3	m4
c1	1.00								
c2	-0.19	1.00							
c3	0.00	0.00	1.00						
c4	0.00	0.00	0.47	1.00					
c5	-0.33	0.58	0.00	-0.31	1.00				
m1	-0.17	-0.30	-0.21	-0.16	-0.17	1.00			
m2	-0.26	-0.45	-0.32	-0.24	-0.26	0.67	1.00		
m3	-0.33	-0.58	-0.41	-0.31	-0.33	0.52	0.77	1.00	
m4	-0.33	-0.19	-0.41	-0.31	-0.33	-0.17	0.26	0.56	1.00

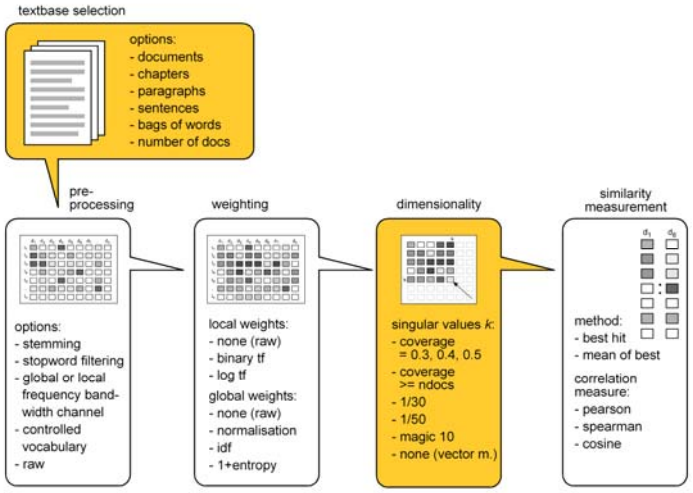
reduced

- based on $M_2 = TS_2D'$
- Pearson Correlation over document vectors

	c1	c2	c3	c4	c5	m1	m2	m3	m4
c1	1.00								
c2	0.91	1.00							
c3	1.00	0.91	1.00						
c4	1.00	0.88	1.00	1.00					
c5	0.85	0.99	0.85	0.81	1.00				
m1	-0.85	-0.56	-0.85	-0.88	-0.45	1.00			
m2	-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1.00		
m3	-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1.00	1.00	
m4	-0.81	-0.50	-0.81	-0.84	-0.37	1.00	1.00	1.00	1.00

<7>

LSA Process & Driving Parameters



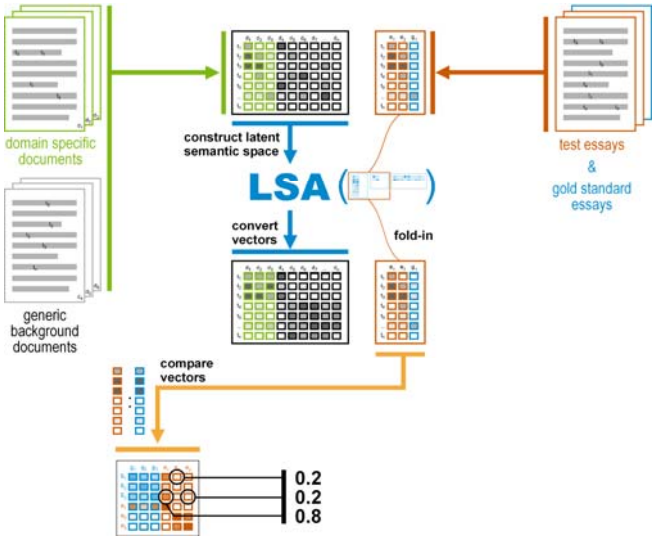
<8>

The Package

- Available via CRAN, e.g.:
  http://cran.at.r-project.org/src/contrib/Descriptions/lsa.html

- Higher-level Abstraction to Ease Use
  - Five core methods:
    `textmatrix() / query()`
    `lsa()`
    `fold_in()`
    `as.textmatrix()`
  - Supporting methods for term weighting, dimensionality calculation, correlation measurement, triple binding

Demo I: Essay Scoring with LSA



Essay Scoring (Code)

```
library( "lsa“ )                        # load package

# load training texts
trm = textmatrix( "trainingtexts/“ )
trm = lw_bintf( trm ) * gw_idf( trm )  # weighting
space = lsa( trm )                      # create an LSA space

# fold-in essays to be tested (including gold standard text)
tem = textmatrix( "testessays/", vocabulary=rownames(trm) )
tem_red = fold_in( tem, space )

# score an essay by comparing with
# gold standard text (very simple method!)
cor( tem_red[,"goldstandard.txt"], tem_red[,"E1.txt"] )
=> 0.7
```