An Introduction to Latent Semantic Analysis

Melanie Martin

October 14, 2002

NMSU CS AI Seminar

Acknowledgements

- Peter Foltz for conversations, teaching me how to use LSA, pointing me to the important work in the field. Thanks!!!
- ARL Grant for supporting this work

Outline

- The Problem
- Some History
- LSA
- A Small Example
- Summary
- Applications: October 28th, by Peter Foltz

The Problem

- Information Retrieval in the 1980s
- Given a collection of documents: retrieve documents that are relevant to a given query
- Match terms in documents to terms in query
- Vector space method

The Problem

- The vector space method
  - term (rows) by document (columns) matrix, based on occurrence
  - translate into vectors in a vector space
    - one vector for each document
  - cosine to measure distance between vectors (documents)
    - small angle = large cosine = similar
    - large angle = small cosine = dissimilar

The Problem

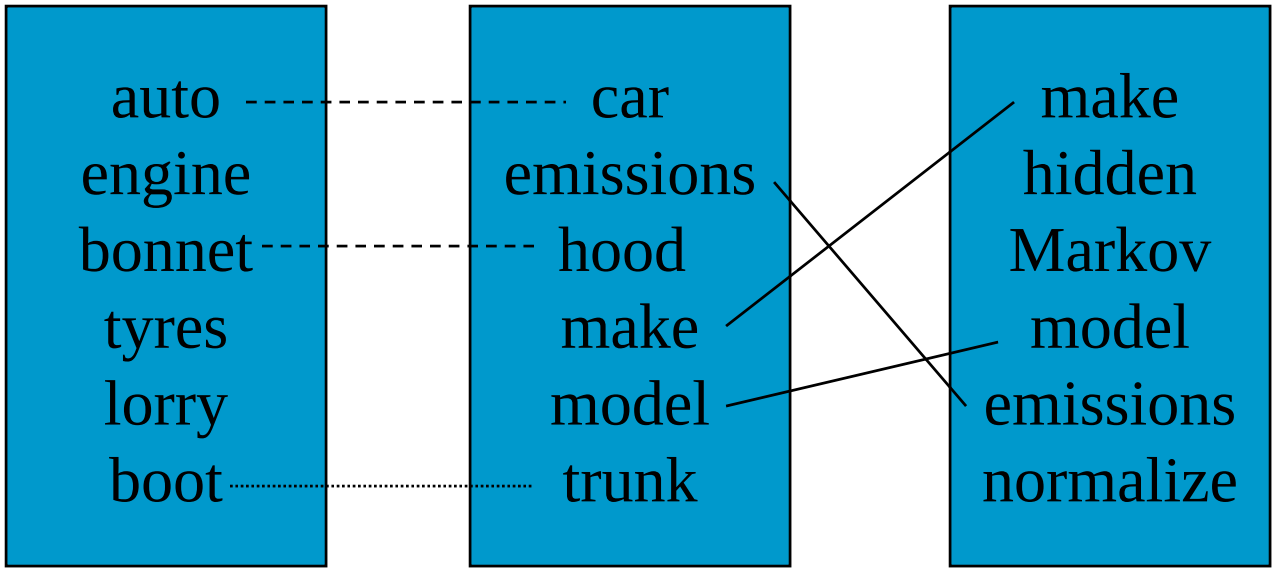
- A quick diversion
- Standard measures in IR
  - Precision: portion of selected items that the system got right
  - Recall: portion of the target items that the system selected

The Problem

- Two problems that arose using the vector space model:
  - synonymy: many ways to refer to the same object, e.g. car and automobile
    - leads to poor recall
  - polysemy: most words have more than one distinct meaning, e.g.model, python, chip
    - leads to poor precision

The Problem

- Example: Vector Space Model
  - (from Lillian Lee)



Synonymy

Will have small cosine

but are related

Polysemy

Will have large cosine

but not truly related

The Problem

- Latent Semantic Indexing was proposed to address these two problems with the vector space model for Information Retrieval

Some History

- Latent Semantic Indexing was developed at Bellcore (now Telcordia) in the late 1980s (1988). It was patented in 1989.
- http://lsi.argreenhouse.com/lsi/LSI.html

Some History

- The first papers about LSI:
  - Dumais, S. T., Furnas, G. W., Landauer, T. K. and Deerwester, S. (1988), "Using latent semantic analysis to improve information retrieval." In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, 281-285.
  - Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R.A. (1990) "Indexing by latent semantic analysis." Journal of the Society for Information Science, 41(6), 391-407.
  - Foltz, P. W. (1990) "Using Latent Semantic Indexing for Information Filtering". In R. B. Allen (Ed.) Proceedings of the Conference on Office Information Systems, Cambridge, MA, 40-47.

LSA

- But first:
- What is the difference between LSI and LSA???
  - LSI refers to using it for indexing or information retrieval.
  - LSA refers to everything else.

LSA

- Idea (Deerwester et al):

"We would like a representation in which a set of terms, which by itself is incomplete and unreliable evidence of the relevance of a given document, is replaced by some other set of entities which are more reliable indicants. We take advantage of the implicit higher-order (or latent) structure in the association of terms and documents to reveal such relationships."

LSA

- Implementation: four basic steps
  - term by document matrix (more generally term by context) tend to be sparce
  - convert matrix entries to weights, typically:
    - L(i,j) * G(i): local and global
    - a_ij -> log(freq(a_ij)) divided by entropy for row (-sum (p logp), over p: entries in the row)
      - weight directly by estimated importance in passage
      - weight inversely by degree to which knowing word occurred provides information about the passage it appeared in

LSA

- Four basic steps
  - Rank-reduced Singular Value Decomposition (SVD) performed on matrix
    - all but the k highest singular values are set to 0
    - produces k-dimensional approximation of the original matrix (in least-squares sense)
    - this is the “semantic space”
  - Compute similarities between entities in semantic space (usually with cosine)

LSA

- SVD
  - unique mathematical decomposition of a matrix into the product of three matrices:
    - two with orthonormal columns
    - one with singular values on the diagonal
  - tool for dimension reduction
  - similarity measure based on co-occurrence
  - finds optimal projection into low-dimensional space

LSA

- SVD
  - can be viewed as a method for rotating the axes in n-dimensional space, so that the first axis runs along the direction of the largest variation among the documents
    - the second dimension runs along the direction with the second largest variation
    - and so on
  - generalized least-squares method

A Small Example

- To see how this works let's look at a small example
- This example is taken from:
  Deerwester, S.,Dumais, S.T., Landauer, T.K.,Furnas, G.W. and Harshman, R.A. (1990). "Indexing by latent semantic analysis." *Journal of the Society for Information Science, 41(6)*, 391-407.
- Slides are from a presentation by Tom Landauer and Peter Foltz

A Small Example

Technical Memo Titles

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*

A Small Example – 2

	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

r (human.user) = -.38 r (human.minors) = -.29

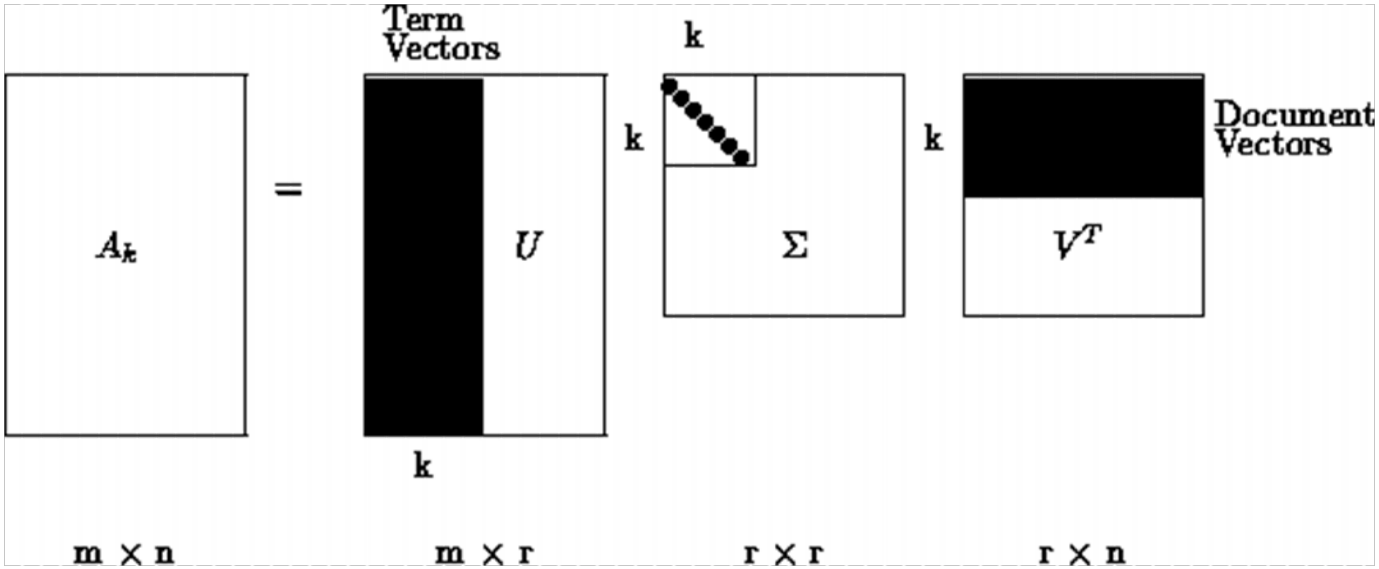
A Small Example – 3

- Singular Value Decomposition

$$\{A\}=\{U\}\{\Sigma\}\{V\}^T$$

- Dimension Reduction

$$\{\tilde{A}\}\sim=\{\tilde{U}\}\{\tilde{\Sigma}\}\{\tilde{V}\}^T$$



A Small Example – 4

- {U} =

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

A Small Example – 5

- $\{\Sigma\} =$

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

A Small Example – 6

- {V} =

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

A Small Example – 7

	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

r (human.user) = .94 r (human.minors) = -.83

A Small Example – 2 reprise

	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

r (human.user) = -.38 r (human.minors) = -.29

Correlation

Raw data

	c1	*c2*	*c3*	*c4*	*c5*	*m 1*	*m 2*	*m 3*
c2	- 019							
c3	0.00	0.00						
c4	0.00	0.00	0.47					
c5	- 033	0.58	0.00	- 031				
m 1	- 017	- 030	- 021	- 016	- 017			
m 2	- 026	- 045	- 032	- 024	- 026	0.67		
m 3	- 033	- 058	- 041	- 031	- 033	0.52	0.77	
m 4	- 033	- 019	- 041	- 031	- 033	- 017	0.26	0.56

0.02	
- 030	**0.44**

Correlations in first-two dimension space

c2	0.91							
c3	1.00	0.91						
c4	1.00	0.88	1.00					
c5	0.85	0.99	0.85	0.81				
m 1	- 085	- 056	- 085	- 088	- 045			
m 2	- 085	- 056	- 085	- 088	- 044	1.00		
m 3	- 085	- 056	- 085	- 088	- 044	1.00	1.00	
m 4	- 081	- 050	- 081	- 084	- 037	1.00	1.00	1.00

0.92	
-0.72	1.00

A Small Example

- A note about notation:
  - Here we called our matrices
    - $\{A\}=\{U\}\{\Sigma\}\{V\}^T$
  - You may also see them called
    - $\{W\}\{S\}\{P\}^T$
    - $\{T\}\{S\}\{D\}^T$
  - The last one is easy to remember
    - T = term
    - S = singular
    - D = document

Summary

- Some Issues
  - SVD Algorithm complexity O(n^2k^3)
    - n = number of terms
    - k = number of dimensions in semantic space (typically small ~50 to 350)
    - for stable document collection, only have to run once
    - dynamic document collections: might need to rerun SVD, but can also “fold in” new documents

Summary

- Some issues
  - Finding optimal dimension for semantic space
    - precision-recall improve as dimension is increased until hits optimal, then slowly decreases until it hits standard vector model
    - run SVD once with big dimension, say k = 1000
      - then can test dimensions <= k
    - in many tasks 150-350 works well, still room for research

Summary

- Some issues
  - SVD assumes normally distributed data
    - term occurrence is not normally distributed
    - matrix entries are weights, not counts, which may be normally distributed even when counts are not

Summary

- Has proved to be a valuable tool in many areas of NLP as well as IR
  - summarization
  - cross-language IR
  - topics segmentation
  - text classification
  - question answering
  - more

Summary

- Ongoing research and extensions include
  - Probabilistic LSA (Hofmann)
  - Iterative Scaling (Ando and Lee)
  - Psychology
    - model of semantic knowledge representation
    - model of semantic word learning

Summary

- That’s the introduction, to find out about applications:
  - Monday, October 28th
  - same time same place
  - Peter Foltz on *Applications of LSA*

Epilogue

- The group at the University of Colorado at Boulder has a web site where you can try out LSA and download papers
  - http://lsa.colorado.edu/
- Papers are also available at:
  - http://lsi.research.telcordia.com/lsi/LSI.html