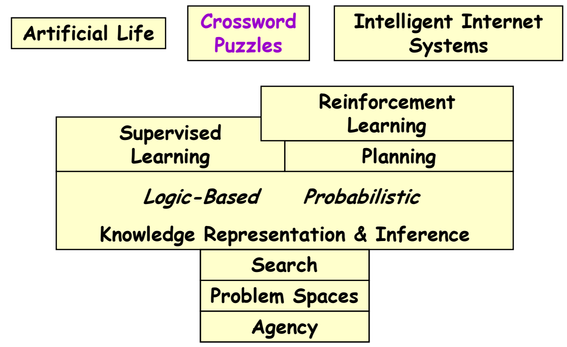
Crosswords, Games, Visualization

CSE 573

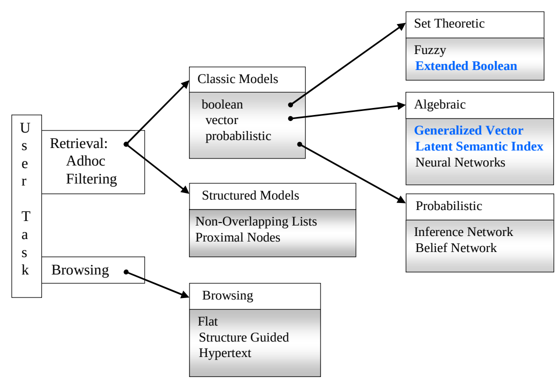
573 Schedule



Logistics

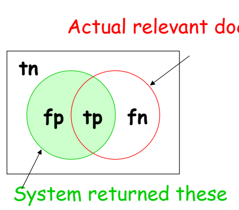
- Information Retrieval Overview
- Crossword & Other Puzzles
- Knowledge Navigator
- Visualization

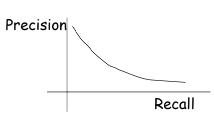
IR Models



Measuring Performance

- Precision $\frac{tp}{tp + fp}$
  Proportion of selected items that are correct
- Recall $\frac{tp}{tp + fn}$
  Proportion of target items that were selected
- Precision-Recall curve
  Shows tradeoff





Precision-recall curves

Easy to get high recall
Just retrieve all docs
Alas… low precision

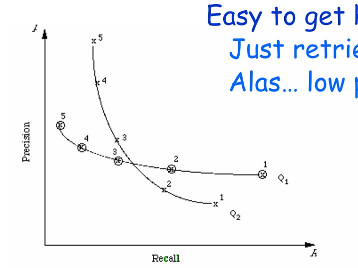


Figure 7.1. The precision-recall curves for two queries. The ordinals indicate the values of the control parameter λ.

The Boolean Model

- Simple model based on set theory
- Queries specified as boolean expressions
  - precise semantics
- Terms are either present or absent.
  - Thus, $w_{ij} \in \{0,1\}$
- Consider
  - $q = ka \wedge (kb \vee \neg kc)$
  - $dnf(q) = (1,1,1) \vee (1,1,0) \vee (1,0,0)$
  - $cc = (1,1,0)$ is a conjunctive component

Drawbacks of the Boolean Model

- Binary decision criteria
  - ***No notion of partial matching***
  - ***No ranking or grading scale***
- Users must write Boolean expression
  - Awkward
  - Often too simplistic
- Hence users get too few or too many documents

Thus ... The Vector Model

- Use of binary weights is too limiting
- [0, 1] term weights are used to compute
  - ***Degree of similarity*** between a query and documents
- Allows ranking of results

Documents as bags of words

a: System and human system engineering testing of EPS
b: A survey of user opinion of computer system response time
c: The EPS user interface management system
d: Human machine interface for ABC computer applications
e: Relation of user perceived response time to error measurement
f: The generation of random, binary, ordered trees
g: The intersection graph of paths in trees
h: Graph minors IV: Widths of trees and well-quasi-ordering
i: Graph minors: A survey

Documents

Terms	a	b	c	d	e	f	g	h	I
Interface	0	0	1	0	0	0	0	0	0
User	0	1	1	0	1	0	0	0	0
System	2	1	1	0	0	0	0	0	0
Human	1	0	0	1	0	0	0	0	0
Computer	0	1	0	1	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	1	0	1	0	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

Terminology: Term Weights

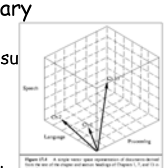
- Not all terms are equally useful for representing the document contents
- Less frequent terms allow identifying a narrower set of documents
  - The *importance* of the index terms is represented by weights associated to them

k_i is an index term
d_j is a document
t is the total number of docs
$K = \{k_1, k_2, \ldots, k_t\}$, the set of all index terms
$w_{ij} >= 0$ is a weight associated with (k_i, d_j)
$w_{ij} = 0$ indicates term missing from doc
$vec(d_j) = (w_{1j}, w_{2j}, \ldots, w_{tj})$ is a weighted vector associated with the document d_j (or query q)

The Vector Model Definitions

- Documents/Queries modeled as bags of words
  - Represented as vectors over keyword sp
  - $vec(dj) = (w1j, w2j, \ldots, wtj)$
  - $vec(q) = (w1q, w2q, \ldots, wtq)$
    - $wij > 0$ whenever $ki \in dj$
    - $wiq >= 0$ associated with the pair $(ki,$
  - To each term ki is associated a unitary vector $vec(i)$
    - Unitary vectors $vec(i)$ and $vec(j)$ are assu orthonormal
    - What does this mean?
      - » Is this Reasonable??????
- t unitary vectors $vec(i)$ form
  - an orthonormal basis for a t-dimensional space

- Each vector holds a place for every term in the collection
- Therefore, most vectors are sparse



Vector Space Example

a: System and human system engineering testing of EPS
b: A survey of user opinion of computer system response time
c: The EPS user interface management system
d: Human machine interface for ABC computer applications
e: Relation of user perceived response time to error measurement
f: The generation of random, binary, ordered trees
g: The intersection graph of paths in trees
h: Graph minors IV: Widths of trees and well-quasi-ordering
i: Graph minors: A survey

Documents

	a	b	c	d	e	f	g	h	I
Interface	0	0	1	0	0	0	0	0	0
User	0	1	1	0	1	0	0	0	0
System	2	1	1	0	0	0	0	0	0
Human	1	0	0	1	0	0	0	0	0
Computer	0	1	0	1	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	1	0	1	0	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

Similarity Function

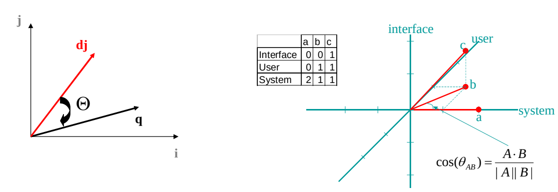
The similarity or closeness of a document
$d = (w_1, \ldots, w_i, \ldots, w_n)$
with respect to a query (or another document)
$q = (q_1, \ldots, q_i, \ldots, q_n)$
is computed using a similarity (distance) function.
Many similarity functions exist

Euclidian Distance

- Given two document vectors d1 and d2

$$Dist(d1, d2) = \sqrt{\sum_i (wi1 - wi2)^2}$$

Cosine metric



$Sim(q,d_j) = \cos(\Theta)$
$= [vec(d_j) \bullet vec(q)] / |d_j| * |q|$
$= [\Sigma\ w_{ij} * w_{iq}] / |d_j| * |q|$

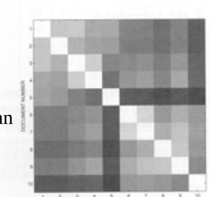
$0 <= sim(q,d_j) <= 1$ (Since $w_{ij} > 0$ and $w_{iq} > 0$)

Retrieves docs even if only partial match to query

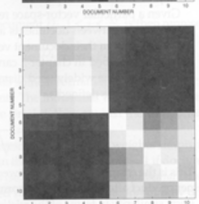
	t1	t2	t3	t4	t5	t6
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

t1= database
t2=SQL
t3=index
t4=regression
t5=likelihood
t6=linear

Eucledian



Cosine



Comparison of Eucledian and Cosine distance metrics

Term Weights in the Vector Model

$Sim(q,d_j) = [\Sigma\ w_{ij} * w_{iq}] / |d_j| * |q|$

How to compute the weights *wij* and *wiq* ?

Simple frequencies favor common words

E.g. Query: The Computer Tomography

A good weight must account for 2 effects:

Intra-document contents (similarity)
tf factor, the ***term frequency*** within a doc

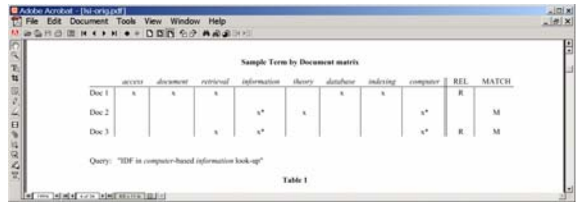
Inter-document separation (dis-similarity)
idf factor, the ***inverse document frequency***

$idf(i) = \log (N/n_i)$

TF-IDF

- Let,
  - N be the total number of docs in the collection
  - ni be the number of docs which contain ki
  - $freq(i,j)$ raw frequency of ki within dj
- A normalized *tf* factor is given by
  - $f(i,j) = freq(i,j) / max(freq(i,j))$
    - where the maximum is computed over all terms which occur within the document dj
- The *idf* factor is computed as
  - $idf(i) = log(N/ni)$
    - the *log* is used to make the values of *tf* and *idf* comparable.
    - Can be interpreted as the *amount of information* associated with the term ki.

Motivating the Need for LSI


Adobe Acrobat - [lsi-orig.pdf]

File Edit Document Tools View Window Help

Sample Term by Document matrix

	access	*document*	*retrieval*	*information*	*theory*	*database*	*indexing*	*computer*	REL	MATCH
Doc 1	x	x	x			x	x		R	
Doc 2				x*	x			x*		M
Doc 3			x	x*				x*	R	M

Query: "IDF in *computer-based information* look-up"

Table 1

-- Relevant docs may not have the query terms
→ but may have many "related" terms
-- Irrelevant docs may *have* the query terms
→ but may not have any "related" terms

Terms and Docs as vectors in "factor" space

In addition to doc-doc similarity,
We can compute term-term distance

Document vector

Term vector

	a	b	c	d	e	f	g	h	I
Interface	0	0	1	0	0	0	0	0	0
User	0	1	1	0	1	0	0	0	0
System	2	1	1	0	0	0	0	0	0
Human	1	0	0	1	0	0	0	0	0
Computer	0	1	0	1	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	1	0	1	0	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

If terms are independent, the T-T similarity matrix would be diagonal
=If it is not diagonal, we can use the correlations to add related terms to the query
=But can also ask the question "Are there independent dimensions which define the space where terms & docs are vectors ?"

Latent Semantic Indexing

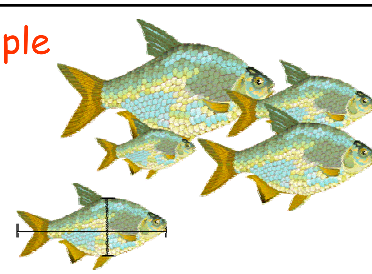
- Creates modified vector space
- Captures transitive co-occurrence information
  - If docs A & B don't share any words, with each other, but both share lots of words with doc C, then A & B will be considered similar
  - Handles polysemy (adam's apple) & synonymy
- Simulates query expansion and document clustering (sort of)

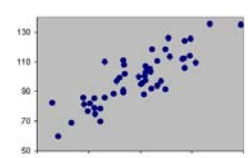
LSI Intuition

- The key idea is to map documents and queries into a lower dimensional space (i.e., composed of higher level concepts which are in fewer number than the index terms)
- Retrieval in this reduced concept space might be superior to retrieval in the space of index terms

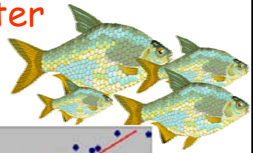
Visual Example

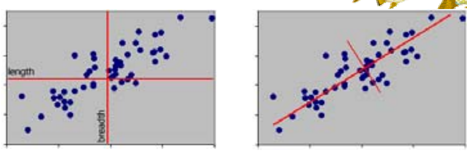
- Classify Fish
  - Length
  - Height





Move Origin to Center



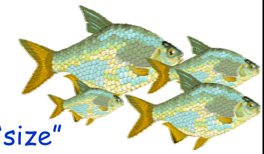


But are these the best axes?

Better if one axis accounts for most data variation

What should we call the red axis?

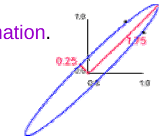
Reduce Dimensions

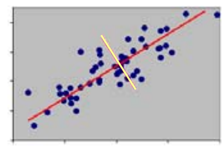


What if we *only* consider "size"

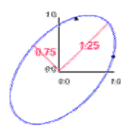
We retain 1.75/2.00 x 100 (87.5%) of the original variation.

Thus, by discarding the yellow axis we lose only 12.5% of the original information.





Not Always Appropriate



Linear Algebra Review

- Let A be a matrix
- X is an Eigenvector of A if

  $A*X = \lambda X$

  $\boxed{A} * \boxed{X} = \lambda \boxed{X}$
- λ is an Eigenvalue
- Transpose:

  $\boxed{A}^T = \boxed{}$

Latent Semantic Indexing Defns

- Let m be the total number of index terms
- Let n be the number of documents
- Let [Aij] be a term-document matrix

  With m rows and n columns

  Entries = weights, wij, associated with the pair [ki,dj]
- The weights can be computed with tf-idf

Singular Value Decomposition

- Factor [Aij] matrix into 3 matrices as follows:
- (Aij) = (U) (S) (V)^t

  (U) is the matrix of eigenvectors derived from (A)(A)^t

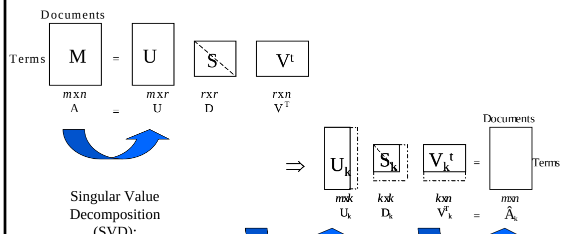
  (V)^t is the matrix of eigenvectors derived from (A)^t(A)

  (S) is an *r x r* diagonal matrix of singular values

  U and V are orthogonal matrices

  - r = min(t,n) that is, the rank of (Aij)
  - Singular values are the positive square roots of the eigen values of (A)(A)^t (also (A)^t(A))

LSI in a Nutshell



Singular Value Decomposition (SVD): Convert term-document matrix into 3 matrices U, S and V

Reduce Dimensionality: Throw out low-order rows and columns

Recreate Matrix: Multiply to produce approximate term-document matrix. Use new matrix to process queries

Example

term	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9
controllability	1	1	0	0	1	0	0	1
observability	1	0	0	0	1	1	0	1
realization	1	0	1	0	1	0	1	0
feedback	0	1	0	0	0	1	0	0
controller	0	1	0	0	1	1	0	0
observer	0	1	1	0	1	1	0	0
transfer function	0	0	0	0	1	1	0	0
polynomial	0	0	0	0	1	0	1	0
matrices	0	0	0	0	1	0	1	1

U (9x7) =

0.3996	-0.1037	0.5606	-0.3717	-0.3919	-0.3482	0.1029
0.4180	-0.0641	0.4878	0.1566	0.5771	0.1981	-0.1094
0.3464	-0.4422	-0.3997	-0.5142	0.2787	0.0102	-0.2857
0.1888	0.4615	0.0049	-0.0279	-0.2087	0.4193	-0.6629
0.3602	0.3776	-0.0914	0.1596	-0.2045	-0.3701	-0.1023
0.4075	0.3622	-0.3657	-0.2684	-0.0174	0.2711	0.5676
0.2750	0.1667	-0.1303	0.4376	0.3844	-0.3066	0.1230
0.2259	-0.3096	-0.3579	0.3127	-0.2406	-0.3122	-0.2611
0.2958	-0.4232	0.0277	0.4305	-0.3800	0.5114	0.2010

S (7x7) =

3.9901	0	0	0	0	0	0
0	2.2813	0	0	0	0	0
0	0	1.6705	0	0	0	0
0	0	0	1.3522	0	0	0
0	0	0	0	1.1818	0	0
0	0	0	0	0	0.6623	0
0	0	0	0	0	0	0.6487

V (7x8) = T

0.2917	-0.2674	0.3883	-0.5393	0.3926	-0.2112	-0.4505
0.3399	0.4811	0.0649	-0.3760	-0.6959	-0.0421	-0.1462
0.1889	-0.0351	-0.4582	-0.5788	0.2211	0.4247	0.4346
-0.0000	-0.0000	-0.0000	-0.0000	0.0000	-0.0000	0.0000
0.6838	-0.1913	-0.1609	0.2535	0.0050	-0.5229	0.3636
0.4134	0.5716	-0.0566	0.3383	0.4493	0.3198	-0.2839
0.2176	-0.5151	-0.4369	0.1694	-0.2893	0.3161	-0.5330
0.2791	-0.2591	0.6442	0.1593	-0.1648	0.5455	0.2998

This happens to be a rank-7 matrix -so only 7 dimensions required

Singular values = Sqrt of Eigen values of AA^T

Now to Reduce Dimensions…

- In the matrix (S), select *k* largest singular values
- Keep the corresponding columns in (U) and $(V)^t$
- The resultant matrix is called $(M)_k$ and is given by
  $(M)_k = (U)_k (S)_k (V)^t_k$
  where k, k < r, is the dimensionality of the concept space
- The parameter k should be
  large enough to allow fitting the characteristics of the data
  small enough to filter out the non-relevant representational details

The classic over-fitting issue

Formally, this will be the rank-k (2) matrix that is closest to M in the matrix norm sense

U (9x7) =

0.3996	-0.1037	0.5606	-0.3717	-0.3919	-0.3482	0.10
0.4180	-0.0641	0.4878	0.1566	0.5771	0.1981	-0.1094
0.3464	-0.4422	-0.3997	-0.5142	0.2787	0.0102	-0.2857
0.1888	0.4615	0.0049	-0.0279	-0.2087	0.4193	-0.6629
0.3602	0.3776	-0.0914	0.1596	-0.2045	-0.3701	-0.1023
0.4075	0.3622	-0.3657	-0.2684	-0.0174	0.2711	0.5676
0.2750	0.1667	-0.1303	0.4376	0.3844	-0.3066	0.1230
0.2259	-0.3096	-0.3579	0.3127	-0.2406	-0.3122	-0.2611
0.2958	-0.4232	0.0277	0.4305	-0.3800	0.5114	0.2010

S (7x7) =

3.9901	0	0	0	0	0	0
0	2.2813	0	0	0	0	0
0	0	1.6705	0	0	0	0
0	0	0	1.3522	0	0	0
0	0	0	0	1.1818	0	0
0	0	0	0	0	0.6623	0
0	0	0	0	0	0	0.6487

V (7x8) =

0.2917	-0.2674	0.3883	-0.5393	0.3926	-0.2112	-0.4505
0.3399	0.4811	0.0649	-0.3760	-0.6959	-0.0421	-0.1462
0.1889	-0.0351	-0.4582	-0.5788	0.2211	0.4247	0.4346
-0.0000	-0.0000	-0.0000	-0.0000	0.0000	-0.0000	0.0000
0.6838	-0.1913	-0.1609	0.2535	0.0050	-0.5229	0.3636
0.4134	0.5716	-0.0566	0.3383	0.4493	0.3198	-0.2839
0.2176	-0.5151	-0.4369	0.1694	-0.2893	0.3161	-0.5330
0.2791	-0.2591	0.6442	0.1593	-0.1648	0.5455	0.2998

U2 (9x2) =

0.3996	-0.1037
0.4180	-0.0641
0.3464	-0.4422
0.1888	0.4615
0.3602	0.3776
0.4075	0.3622
0.2750	0.1667
0.2259	-0.3096
0.2958	-0.4232

S2 (2x2) =

3.9901	0
0	2.2813

V2 (8x2) = T

0.2917	-0.2674
0.3399	0.4811
0.1889	-0.0351
-0.0000	-0.0000
0.6838	-0.1913
0.4134	0.5716
0.2176	-0.5151
0.2791	-0.2591

U2*S2*V2 will be a 9x8 matrix That approximates original matrix

What should be the value of k?

5 components ignored

K=2

$USV^T = U_7S_7V_7^T$

term	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9
controllability	1	1	0	0	1	0	0	1
observability	1	0	0	0	1	1	0	1
realization	1	0	1	0	1	0	1	0
feedback	0	1	0	0	0	1	0	0
controller	0	1	0	0	1	1	0	0
observer	0	1	1	0	1	1	0	0
transfer function	0	0	0	0	1	1	0	0
polynomial	0	0	0	0	1	0	1	0
matrices	0	0	0	0	1	0	1	1

K=4

3 components ignored

K=6

One component ignored

$U_2S_2V_2^T$

0.52835834	0.42813724	0.30949408	0.0	1.1355368	0.5239192	0.46880865	0.5063048
0.5256176	0.49655432	0.3201918	0.0	1.1684579	0.6059082	0.4382505	0.50338876
0.6729299	-0.015529543	0.29650056	0.0	1.1381099	-0.0052356124	0.82038856	0.6471
-0.0617774	0.76256883	0.10535021	0.0	0.3137232	0.9132189	-0.37838274	-0.06253
0.18889774	0.90294445	0.24125765	0.0	0.81799114	1.0865396	-0.1309748	0.17793834
0.25334513	0.95019233	0.27814224	0.0	0.9537667	1.1444798	-0.071810216	0.2397161
0.21838559	0.55592346	0.19392742	0.0	0.6775683	0.6709899	0.042878807	0.2077163
0.4517898	-0.033422917	0.19505836	0.0	0.75146574	-0.031091988	0.55994695	0.4345
0.60244554	-0.06330189	0.25684044	0.0	0.99175954	-0.06392482	0.75412846	0.5795

$U_4S_4V_4^T$

1.1630535	0.67789733	0.17131016	0.0	0.85744447	0.30088043	-0.025483057	1.0295205
0.7278324	0.46981966	-0.1757451	0.0	1.0910251	0.6314231	0.11810507	1.0620605
0.78863835	0.20257005	1.0048805	0.0	1.0692837	-0.20266426	0.9943222	0.106248446
-0.03825318	0.7772852	0.12343567	0.0	0.30284256	0.89999276	-0.3883498	-0.06326774
0.013223715	0.8118903	0.18630582	0.0	0.8972661	1.1681904	-0.027708884	0.11395822
0.21186034	1.0470067	0.76812166	0.0	0.960058	1.0562774	0.1336124	-0.2116417
-0.18525022	0.31930918	-0.048827052	0.0	0.8625925	0.8834896	0.23821498	0.1617572
-0.008397698	-0.23121	0.2242676	0.0	0.9548515	0.14579195	0.89278513	0.1167786
0.30647483	-0.27917668	-0.101294056	0.0	1.1318822	0.13038804	0.83252335	0.70210195

$U_6S_6V_6^T$

1.0299273	1.0099105	-0.029033005	0.0	0.9757162	0.019038305	0.035608776	0.98004794
0.96788234	-0.010319378	0.030770123	0.0	1.0258299	0.9798115	-0.03772955	1.0212346
0.9165214	-0.026921304	1.0805727	0.0	1.0673982	-0.052518982	0.9011715	0.055653755
-0.19373542	0.9372319	0.1868434	0.0	0.15639876	0.87798584	-0.22921464	0.12886547
-0.029890355	0.9903935	0.028769515	0.0	1.0242295	0.98121595	-0.03527296	0.020075336
0.16586632	1.0537577	0.8398298	0.0	0.8660687	1.1044582	0.19631699	-0.11030859
0.035988174	0.01172187	-0.03462495	0.0	0.9710446	1.0226605	0.04260301	-0.023878671
-0.07636017	-0.024632007	0.07358454	0.0	1.0615499	-0.048087567	0.909685	0.050844945
0.05863098	0.019081593	-0.056740552	0.0	0.95253044	0.03693092	1.0695065	0.96087193

Coordinate transformation inherent in LSI

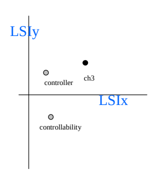
$M = U S V^T$

Mapping of keywords into LSI space is given by US

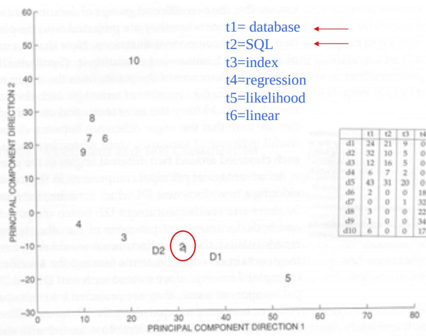
Mapping of a doc d=[w1….wk] into LSI space is given by dUS^{-1}

For k=2, the mapping is:

	LSx	LSy
controllability	1.5944439	-0.2365708
observability	1.6678618	-0.14623132
realization	1.3821706	-1.0087909
feedback	0.7533309	1.05282
controller	1.4372339	0.86141896
observer	1.6259657	0.82628685
Transfer function	1.0972775	0.38029274
polynomial	0.90136355	-0.7062905
matrices	1.1802715	-0.96544623



The base-keywords of The doc are first mapped To LSI keywords and Then differentially weighted By S^{-1}



	t1	t2	t3	t4	t5	t6
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

Figure 14.3 Projected locations of the 10 documents (from table 14.2) in the two dimensional plane spanned by the first two principal components of the document-term matrix M.

Calculating Information Loss

$$\lambda_1, \ldots, \lambda_6 = \{77.4, 69.5, 22.9, 13.5, 12.1, 4.8\}.$$

In agreement with our intuition, most of the variance in the data is captured by the first two principal components. In fact, if we were to retain only these two principal components (as two surrogate terms instead of the six original terms), the fraction of variance that our two-dimensional representation retains is $\boxed{(\lambda_1^2+\lambda_2^2)/\sum_{i=1}^{6}\lambda_i^2 = 0.925}$; i.e., only 7.5% of the information has been lost (in a mean-square sense). If we represent the documents in the new two-dimensional principal component space, the coefficients for each document correspond to the first two columns of the U matrix:

d1	30.8998	-11.4912
d2	30.3131	-10.7801
d3	18.0007	-7.7138
d4	8.3765	-3.5611
d5	52.7057	-20.6051
d6	14.2118	21.8263
d7	10.8052	21.9140
d8	11.5080	28.0101
d9	9.5259	17.7666
d10	19.9219	45.0751

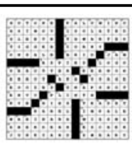
Should clean this up into a slide summarizing the info loss formula

SVD Computation complexity

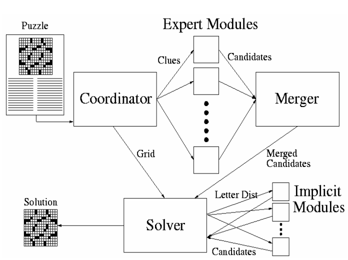
- For an mxn matrix SVD computation is
  - O($km^2n+k'n^3$) complexity
    - k=4 and k'=22 for best algorithms
  - Approximate algorithms that exploit the sparsity of M are available (and being developed)

What LSI can do

- LSI analysis effectively does
  - Dimensionality reduction
  - Noise reduction
  - Exploitation of redundant data
  - Correlation analysis and Query expansion (with related words)
- Any one of the individual effects can be achieved with simpler techniques (see thesaurus construction). But LSI does all of them together.



PROVERB



30 Expert Modules

- Including…
- Partial Match
  - TF/IDF measure
- LSI

PROVERB

- Key ideas

PROVERB

- Weaknesses

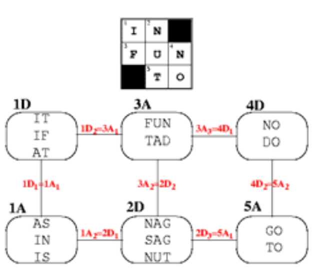
CWDB

- Useful?
  - 94.8% → 27.1%
- Fair?
- Clue transformations
  - Learned

Merging

Modules provide:
- Ordered list
  - <candidate, weight>
- Confidence
- Statistics
  - Scale
  - Scale length
  - Spread

Grid Filling and CSPs



CSPs and IR

Domain from ranked candidate list?

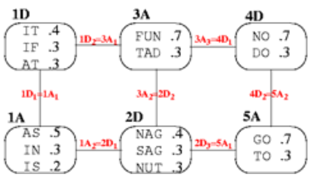
Tortellini topping:

TRATORIA, COUSCOUS, SEMOLINA, PARMESAN, RIGATONI, PLATEFUL, FORDLTDS, SCOTTIES, ASPIRINS, MACARONI, FROSTING, RYEBREAD, STREUSEL, LASAGNAS, GRIFTERS, BAKERIES,… MARINARA, REDMEATS, VESUVIUS, …

Standard recall/precision tradeoff.

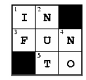
Probabilities to the Rescue?

Annotate domain with the bias.



Solution Probability

Proportional to the product of the probability of the individual choices.



α Pr(**IN**)xPr(**FUN**)xPr(**TO**)x
Pr(**IF**)xPR(**NUT**)xPr(**NO**)
= 0.003969

Can pick sol'n with maximum probability.
Maximizes prob. of whole puzzle correct.
Won't maximize number of words correct.

PROVERB

- Future Work

Trivial Pursuit™

Race around board, answer questions.
Categories: Geography, Entertainment, History, Literature, Science, Sports

Wigwam

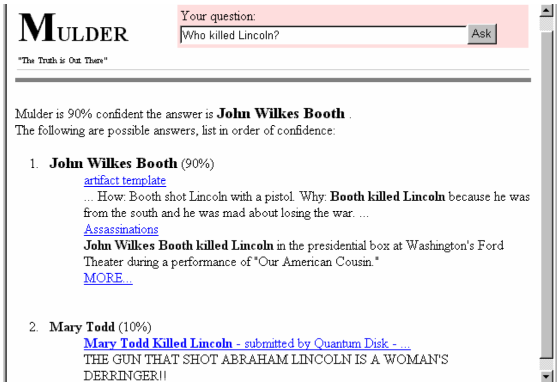
QA via AQUA (Abney et al. 00)
- back off: word match in order helps score.
- "When was Amelia Earhart's last flight?"
  - 1937, 1897 (birth), 1997 (reenactment)
- Named entities only, 100G of web pages

Move selection via MDP (Littman 00)
- Estimate category accuracy.
- Minimize expected turns to finish.

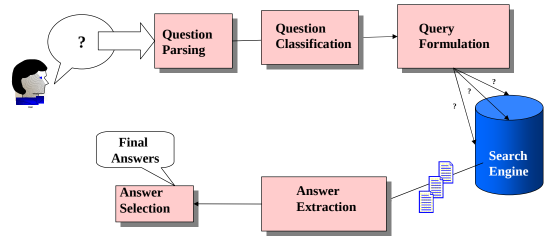
- QA on the Web…

Mulder

- Question Answering System
  User asks Natural Language question: "Who killed Lincoln?"
  Mulder answers: "John Wilkes Booth"
- KB = Web/Search Engines
- Domain-independent
- Fully automated



Architecture



Experimental Methodology

- Idea: In order to answer n questions, how much *user effort* has to be exerted
- Implementation:
  A question is answered if
  - the answer phrases are found in the result pages returned by the service, or
  - they are found in the web pages pointed to by the results.

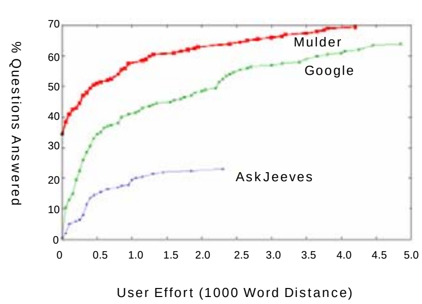
  Bias in favor of Mulder's opponents

Experimental Methodology

- User Effort = Word Distance
  # of words read before answers are encountered

- Google/AskJeeves
  query with the original question

Comparison Results



Knowledge Navigator

Tufte

- Next Slides illustrated from Tufte's book

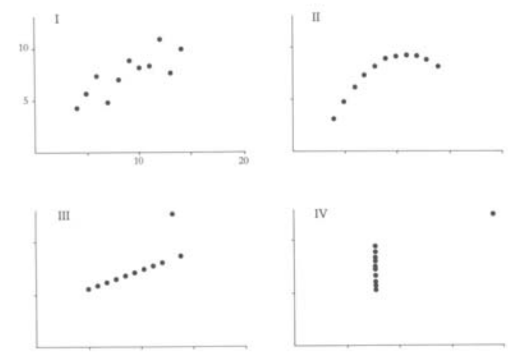
Tabular Data

- Statistically, columns look the same…

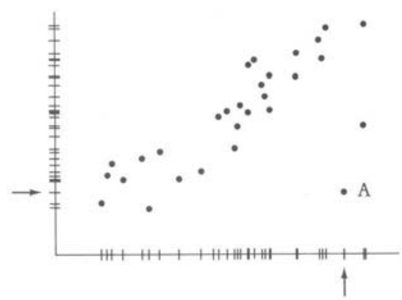
I		II		III		IV	
X	Y	X	Y	X	Y	X	Y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

$N = 11$
mean of X's = 9.0
mean of Y's = 7.5
equation of regression line: $Y = 3 + 0.5X$
standard error of estimate of slope = 0.118
$t = 4.24$
sum of squares $X - \overline{X} = 110.0$
regression sum of squares = 27.50
residual sum of squares of Y = 13.75
correlation coefficient = .82
$r^2 = .67$

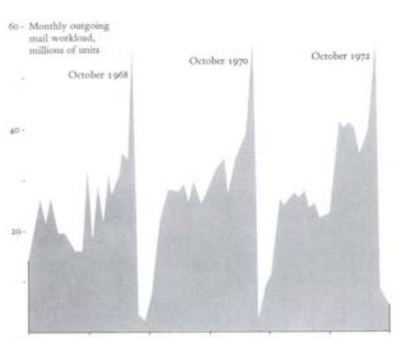
But When Graphed….



Noisy Data?

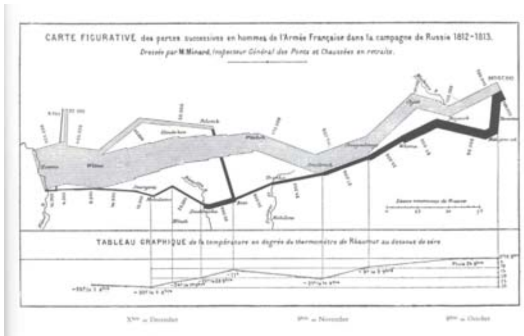


Polictical Control of Economy

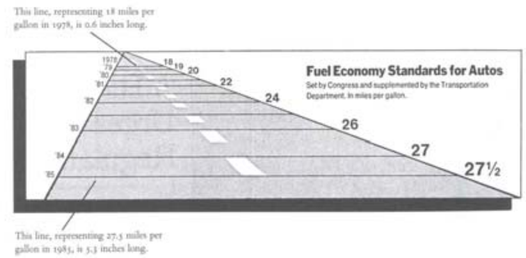


Wine Exports

Napolean



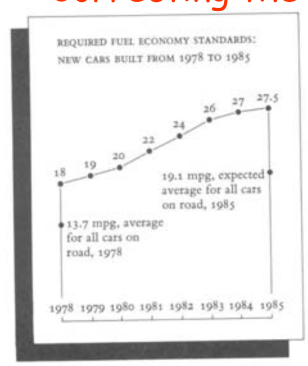
And This Graph?

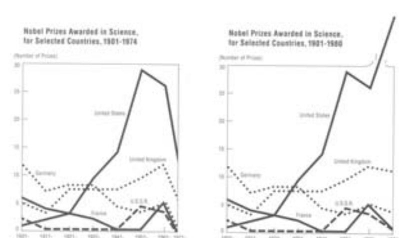


Tufte's Principles

1. The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities themselves
2. Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data.

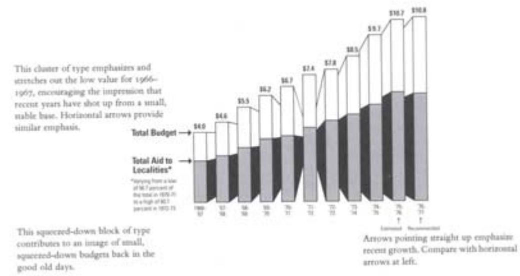
Correcting the Lie





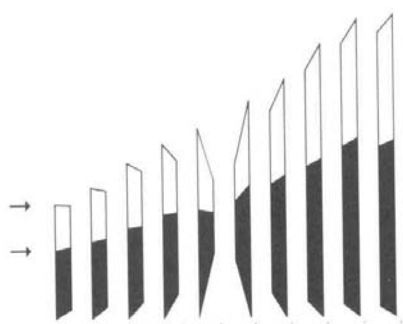
 73

Subtle Distortion



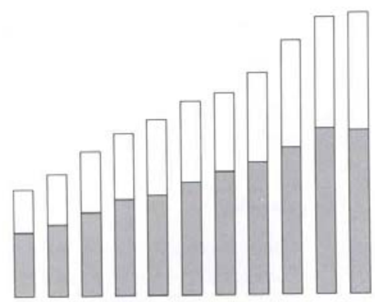
 74

Removing Clutter



 75

Less Busy

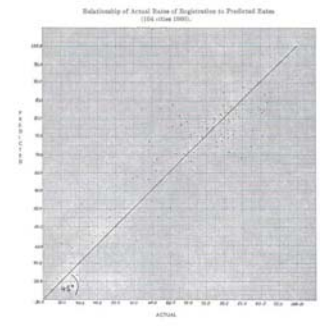


 76

Constant Dollars

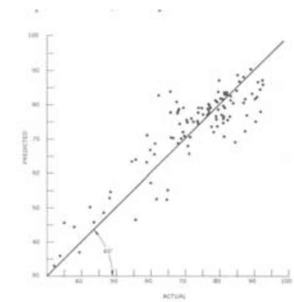
 77

Chart Junk



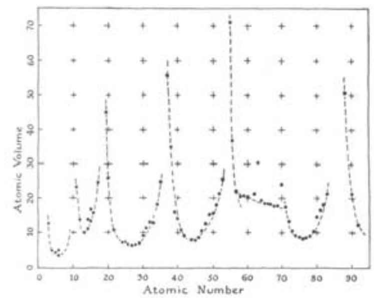
 78

Remove Junk

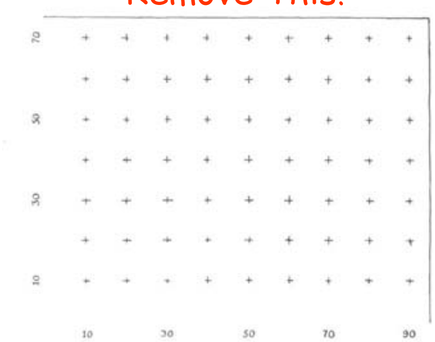


Relationship of Actual Rates of Registration to Predicted Rates (104 cities 1960).

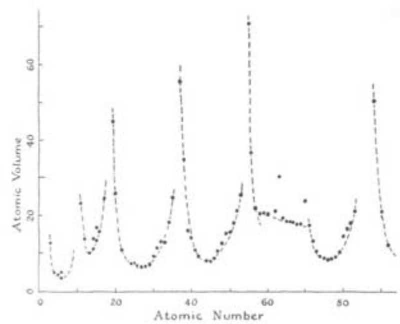
Maximize Data-Ink Ratio



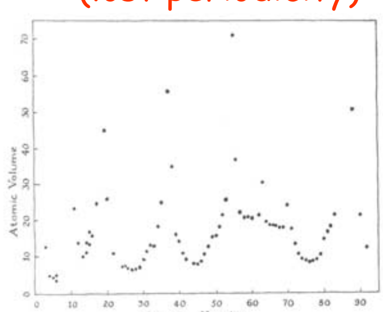
Remove This!



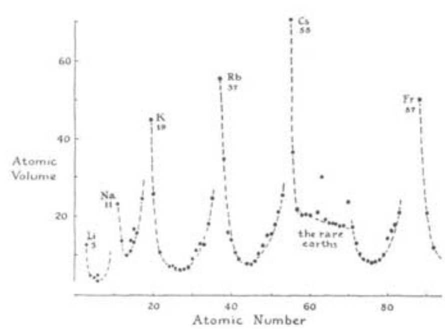
Leaves This



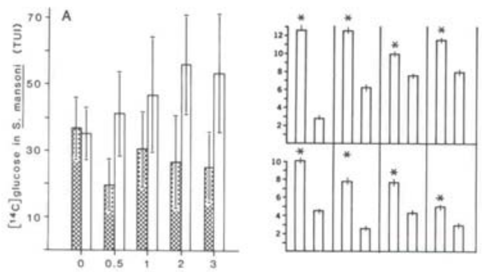
Dropped Too Much (lost periodicity)



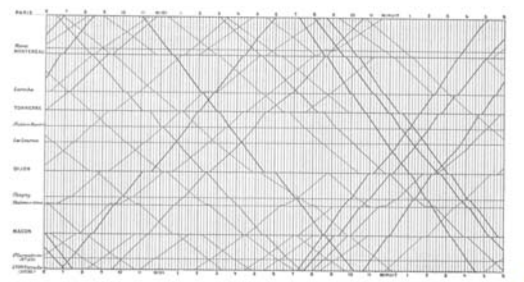
Labeling



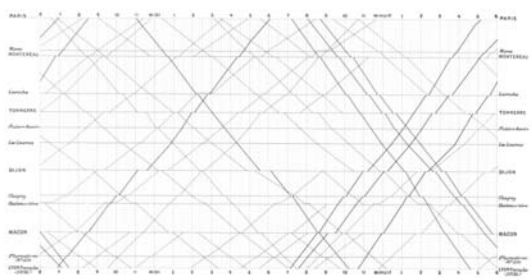
Moire Noise



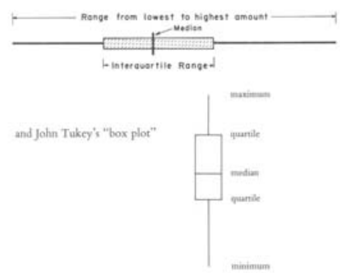
Classic Example



Improved…



DI Ratio

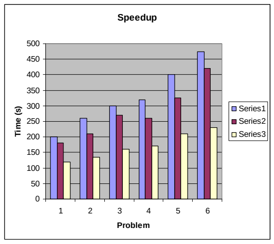


Improved

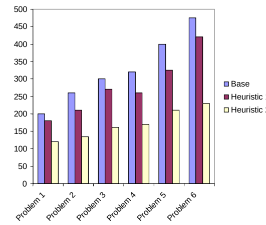
Case Study

	Base Algo	Heuristic 1	Heuristic 2
Problem 1	200	180	120
Problem 2	260	210	135
Problem 3	300	270	160
Problem 4	320	260	170
Problem 5	400	325	210
Problem 6	475	420	230

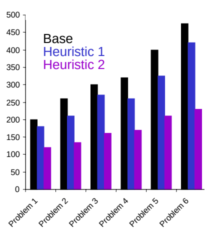
Default Excel Chart



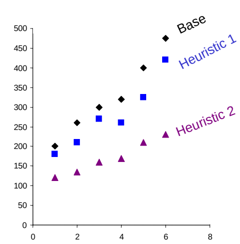
Removing Obvious Chart Junk



Manual Simplification



Scatter Graph



Grand Climax