vector space retrieval

many slides courtesy James Allan@umass Amherst

what is a retrieval model?

- Model is an idealization or abstraction of an actual process
- Mathematical models are used to study the properties of the process, draw conclusions, make predictions
- Conclusions derived from a model depend on whether the model is a good approximation of the actual situation
- Statistical models represent repetitive processes, make predictions about frequencies of interesting events
- Retrieval models can describe the computational process
  - e.g. how documents are ranked
  - Note that how documents or indexes are *stored* is implementation
- Retrieval models can attempt to describe the human process
  - e.g. the information need, interaction
  - Few do so meaningfully
- Retrieval models have an explicit or implicit definition of relevance

retrieval models

- boolean

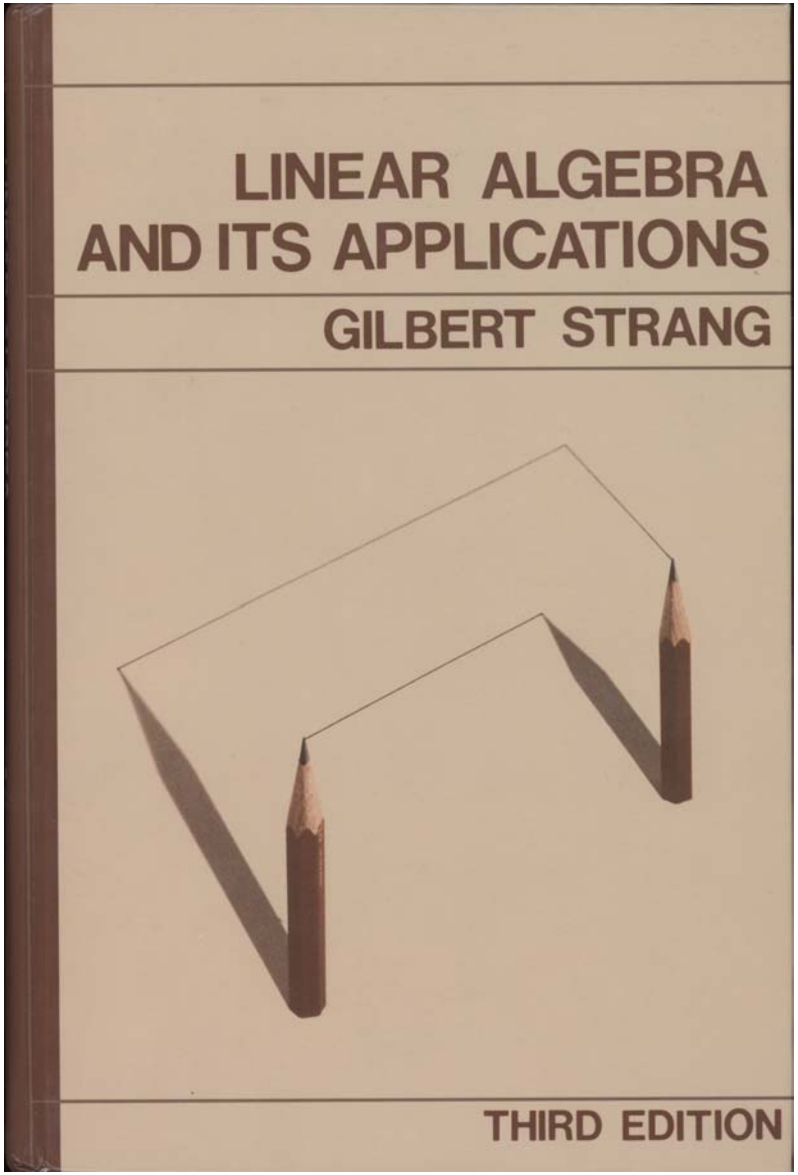
today

- vector space
- latent semnatic indexing
- statistical language
- inference network

outline

- review: geometry, linear algebra
- vector space model
- vector selection
- simmilarity
- weighting schemes
- latent semnatic indexing

linear algebra



vectors

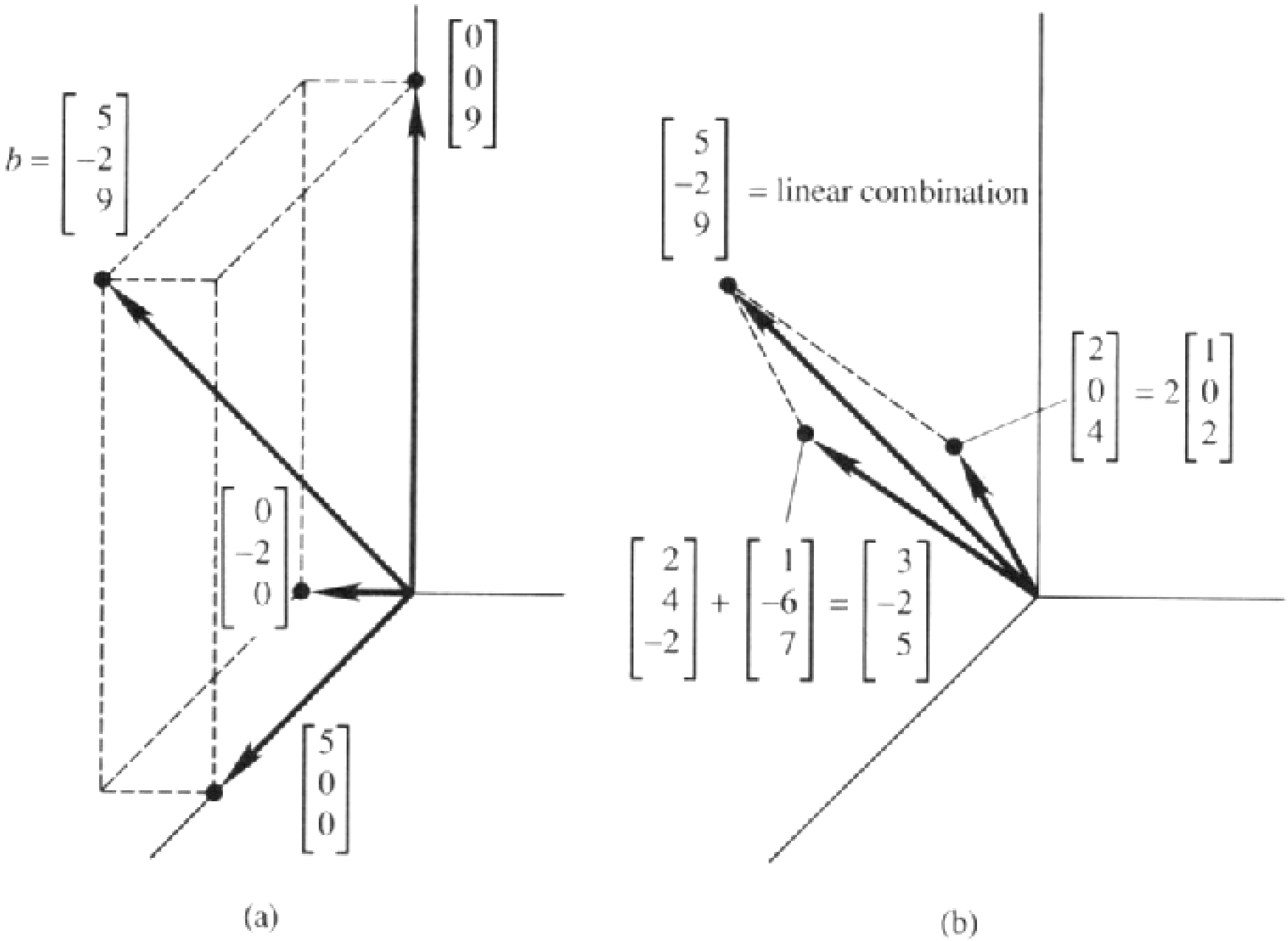
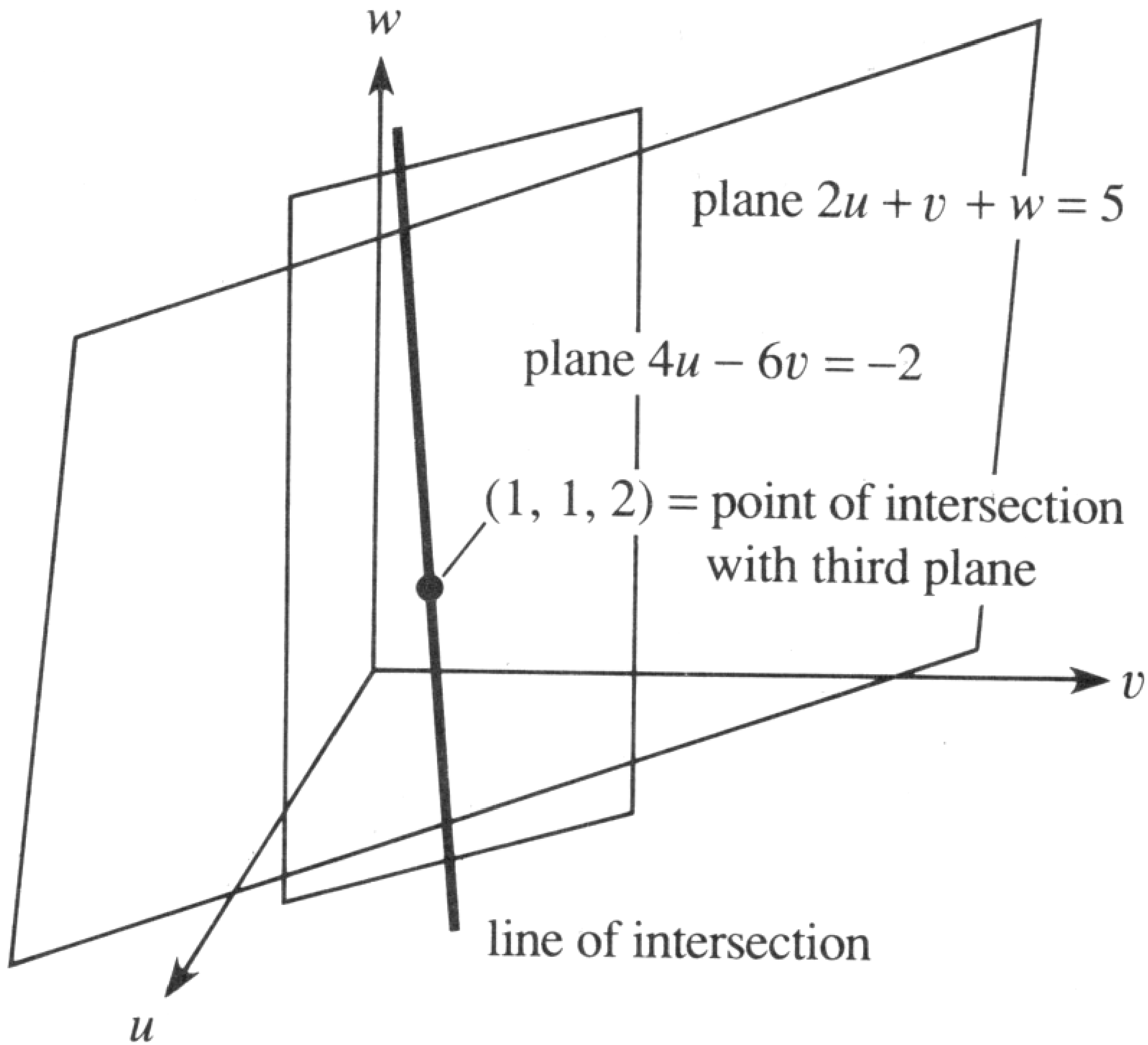


Fig. 1.3. The column picture: linear combination of columns equals b.

subspaces


w
plane $2u + v + w = 5$
plane $4u - 6v = -2$
$(1, 1, 2)$ = point of intersection
with third plane
v
line of intersection
u

linear independence, base, dimmension, rank

- vector $\overline{x}$ is linear dependent of vectors $\overline{y_1}, \overline{y_2}, ..., \overline{y_t}$ if there exists real numbers $c_1, c_2, ..., c_t$ such that

$\overline{x} = c_1\overline{y_1} + c_2\overline{y_2} + ...c_t\overline{y_t}$

- base of a vectorial space = maximal set of linear independent vectors. All bases of a given space have the same dimmension (dimmension of the space)
- rank(A) = maximum number of raws/columns linear independent
- rank(A) = dimenion of the subspace spanned by A

matrix multiplication

$$(AB)_{32} = a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} + a_{34}b_{42}.$$

3 by 4 matrix 4 by 2 matrix 3 by 2 matrix

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix} = \begin{bmatrix} * & * \\ * & * \\ * & (AB)_{32} \end{bmatrix}$$

dot product, norm

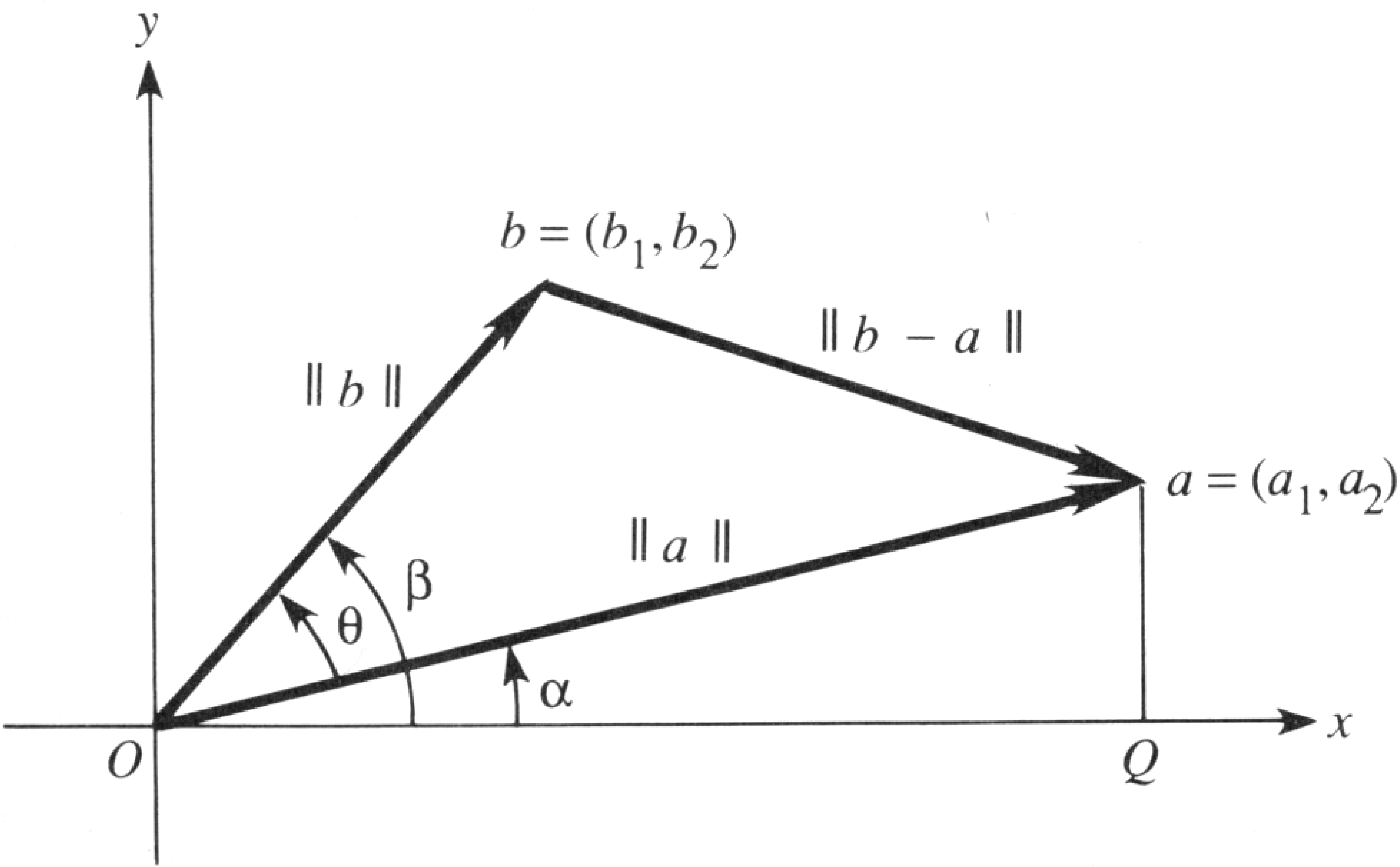
- dot product of 2 same dimension arrays is simply the matrix product with result a real number

- $x = (x_1, x_2, ..., x_n); y = (y_1, y_2, ..., y_n)$ then $< x \cdot y >= x * y^T = \sum_{i=1}^{n} x_i y_i$

- L_2 norm : $||x|| = \sqrt{< x \cdot x >}$

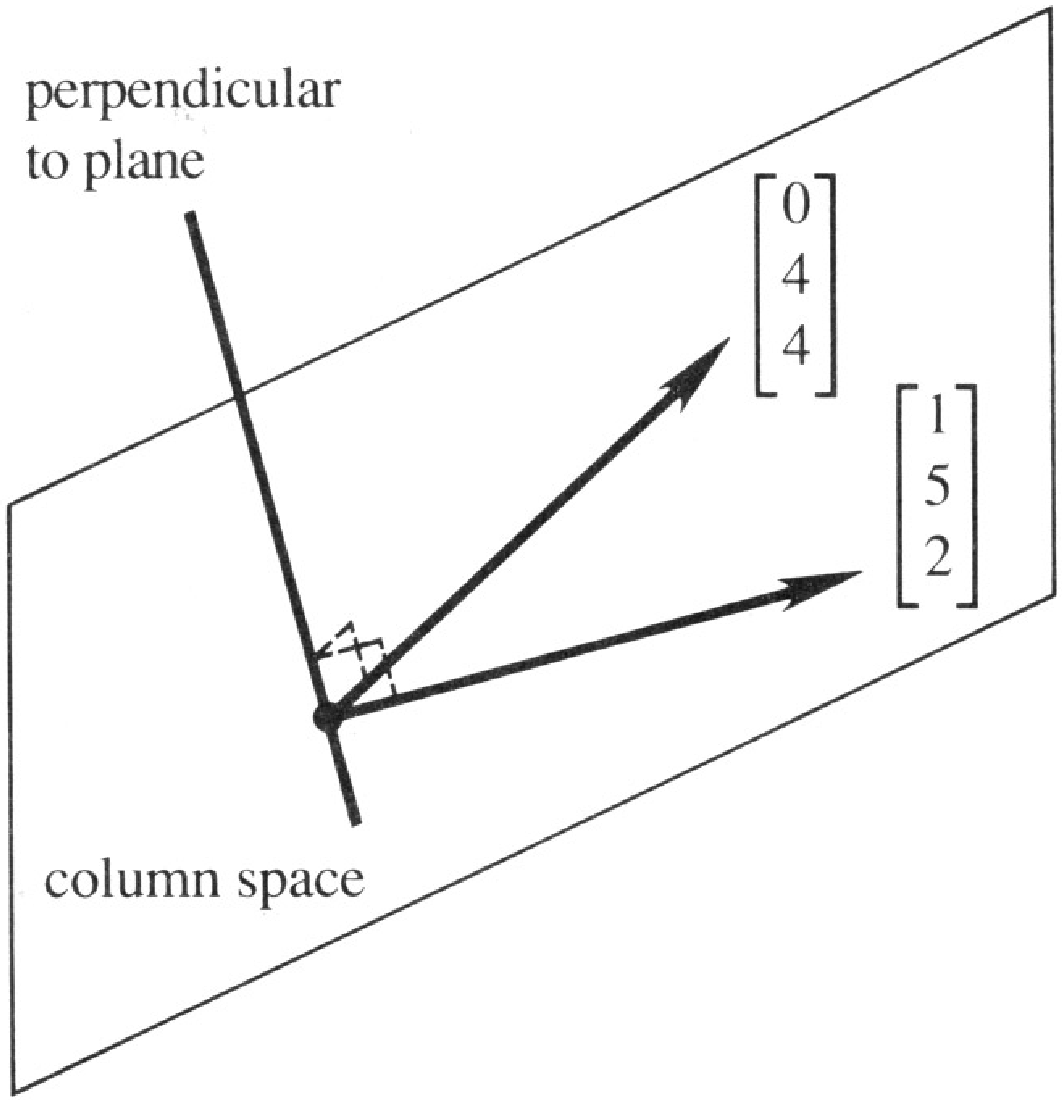
- normalization: $\overline{x} = \frac{x}{||x||}; ||\overline{x}|| = 1$

cosine computation

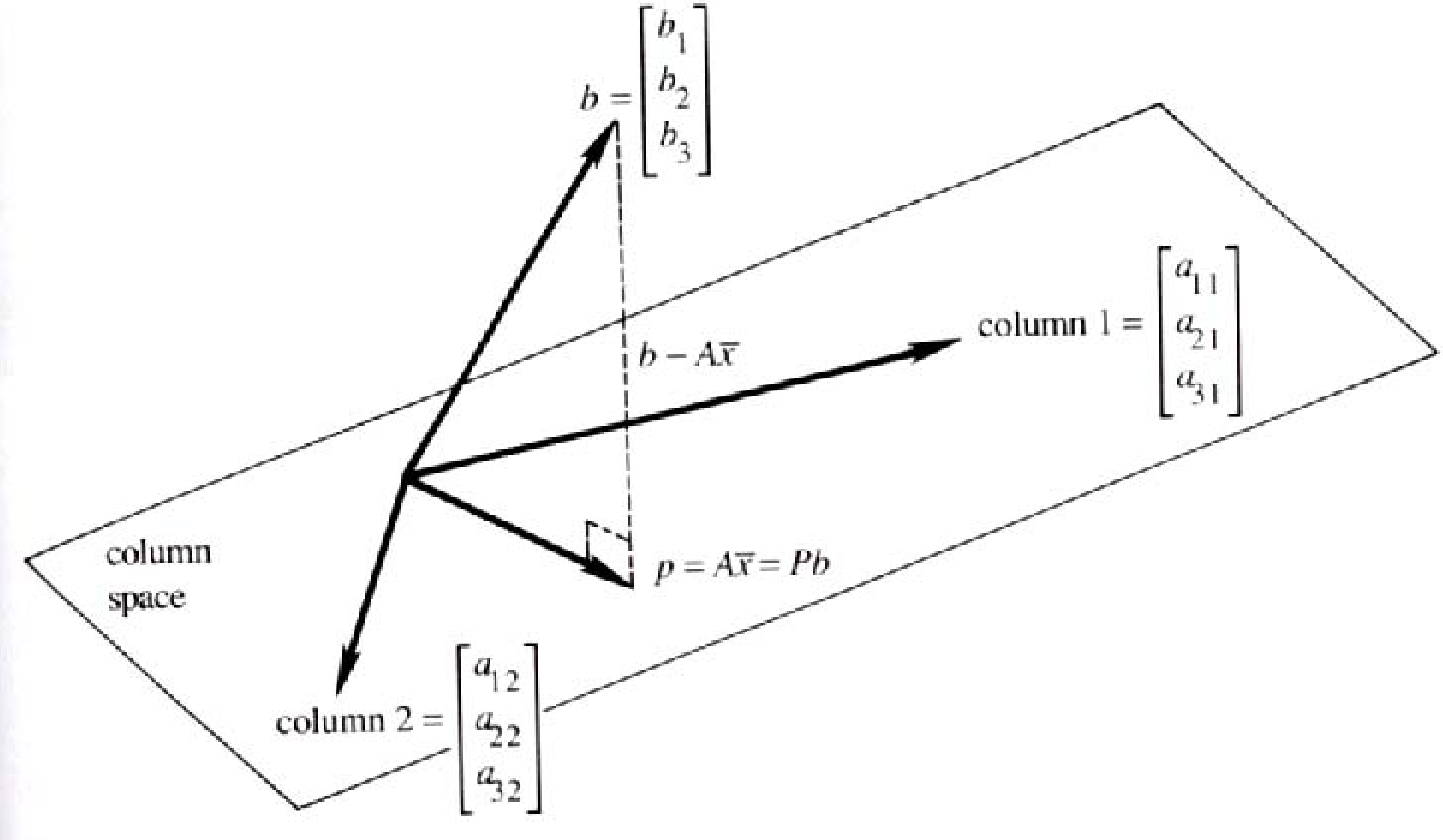


$$\cos(\theta) = \cos(\beta - \alpha) = \cos(\beta)\cos(\alpha) + \sin(\beta)\sin(\alpha)$$

orthogonality



projections



outline

- review: geometry, linear algebra
- vector space model
- vector selection
- simmilarity
- weighting schemes
- latent semnatic indexing

vector space

• represent documents and queries as vectors in the term space

• issue: find the right coefficients (many variants)

• use a geometric similarity measure, often angle-related

• issue: normalization

mapping to vectors

- terms: an axis for every term
  - vectors coresponding to terms are canonical vectors
- documents: sum of the vectors corresponding to terms in the doc
- queries: treated the same as documents

coefficients

- The coefficients (vector lengths, term weights) represent term presence, importance, or “aboutness”
  - Magnitude along each dimension

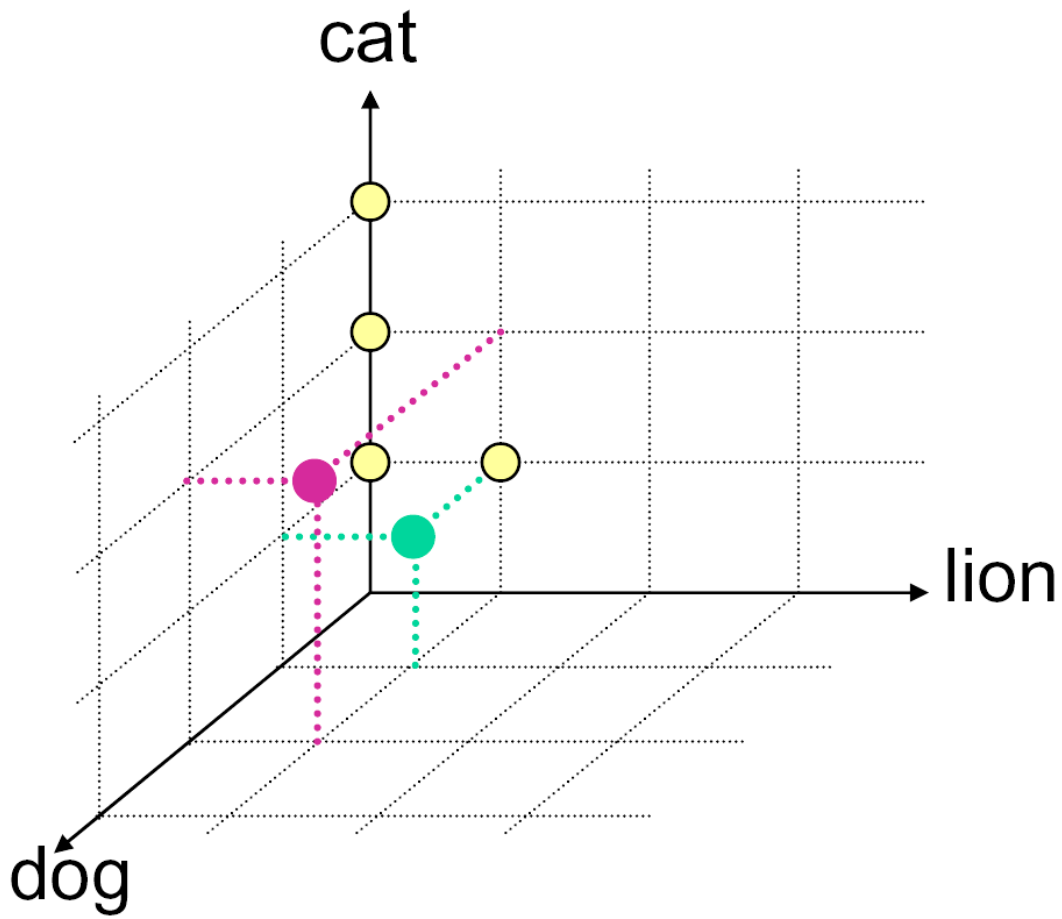
- Model gives no guidance on how to set term weights

- Some common choices:
  - Binary: 1 = term is present, 0 = term not present in document
  - *tf*: The frequency of the term in the document
  - *tf* • *idf*: *idf* indicates the discriminatory power of the term

- Tf·idf is far and away the most common
  - Numerous variations…

example: raw tf weights

- cat
- cat cat
- cat cat cat
- cat lion
- lion cat
- cat lion dog
- cat cat lion dog dog



tf = term frequency

- raw tf (called tf) = count of 'term' in document

- robinson tf (okapi_tf): $okapi\_tf = \frac{tf}{tf + .5 + 1.5\frac{doclen}{avgdoclen}}$

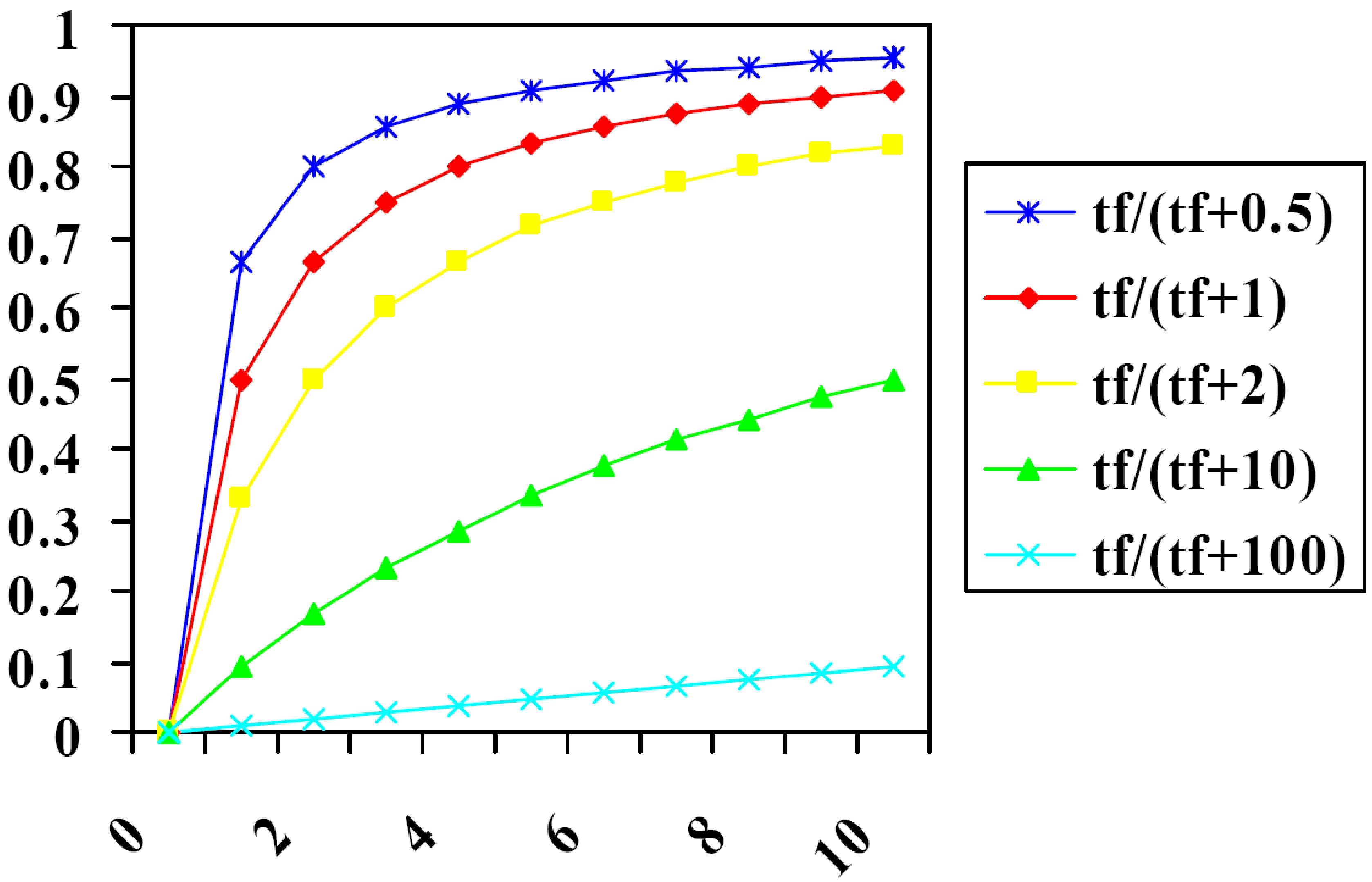
- Based on a set of simple criteria loosely connected to the 2-Poisson model

- Basic formula is tf/(k+tf) where k is a constant (approx. 1-2)

- Document length introduced as a verbosity factor

- many variants

Robertson tf



IDF weights

- Inverse Document Frequency
- used to weight terms based on frequency in the corpus (or language)
- fixed, it can be precomputed for every term
- (basic) $IDF(t) = \log(\frac{N}{N_t})$ where

$N =$ # of docs

$N_t =$ # of docs containing term t

TFIDF

- in fact tf*idf
- the weight on every term is tf(t,d)*idf(t)

Often : IDF= $\log(N/df) + 1$ where N is the number of documents in the collection, df is the number of documents the term occurs in

IDF $= log\frac{1}{p}$, wher p is the term probability

sometimes normalized when in TF.IDF combination

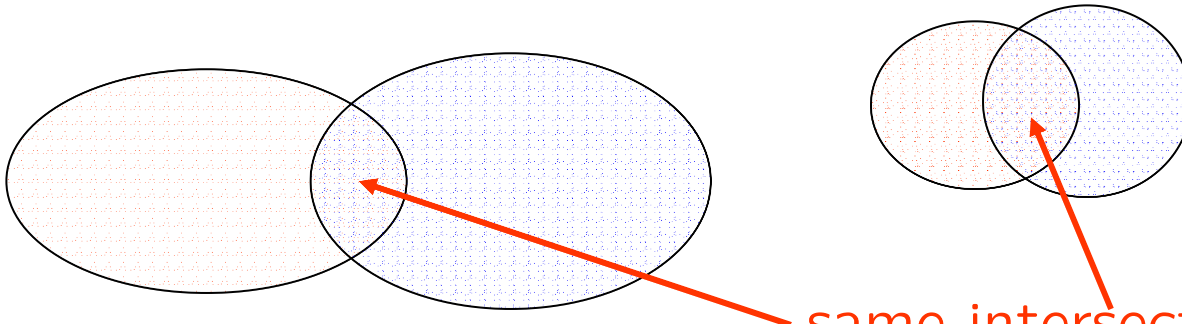
e.g. for INQUERY: $\frac{\log(\frac{N+0.5}{df})}{\log(N+10)}$

- TF and IDF combined using multiplication
- No satisfactory model behind these combinations

outline

- review: geometry, linear algebra
- vector space model
- vector selection
- simmilarity
- weighting schemes
- latent semnatic indexing

similarity, normalized



same intersection area
but different fraction of sets

$$similarity = \frac{|\mathbf{intersection}|}{|\mathbf{set}_1| \cdot |\mathbf{set}_2|}$$

- the size of intersection alone is meaningless
- often divided by sizes of sets
- same for vectors, using norm
  - by normalizing vectors, cosine does not change

common similarity measures

Sim(X,Y)	Binary Term Vectors	Weighted Term Vectors
Inner product	$\lvert X \cap Y \rvert$	$\sum x_i . y_i$
Dice coefficient	$\frac{2 \lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert}$	$\frac{2 \sum x_i . y_i}{\sum x_i^2 + \sum y_i^2}$
Cosine coefficient	$\frac{\lvert X \cap Y \rvert}{\sqrt{\lvert X \rvert} \sqrt{\lvert Y \rvert}}$	$\frac{\sum x_i . y_i}{\sqrt{\sum x_i^2 . \sum y_i^2}}$
Jaccard coefficient	$\frac{\lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert - \lvert X \cap Y \rvert}$	$\frac{\sum x_i . y_i}{\sum x_i^2 + \sum y_i^2 - \sum x_i . y_i}$

similarity: weighted features

T_1 T_2 T_3

$D_1 = 3\ cat + 1\ dog + 4\ lion$
$D_2 = 8\ cat + 2\ dog + 6\ lion$

$D_1 = (3T_1 + 1T_2 + 4T_3)$
$D_2 = (8T_2 + 2T_2 + 6T_3)$

$Q = 2\ dog$

$Q = (0T_1 + 2T_2 + 0T_3)$

Correlated Terms

	Term	cat	dog	lion
T_1	cat	1.00	-0.20	0.50
T_2	dog	-0.20	1.00	-0.40
T_3	lion	0.50	-0.40	1.00

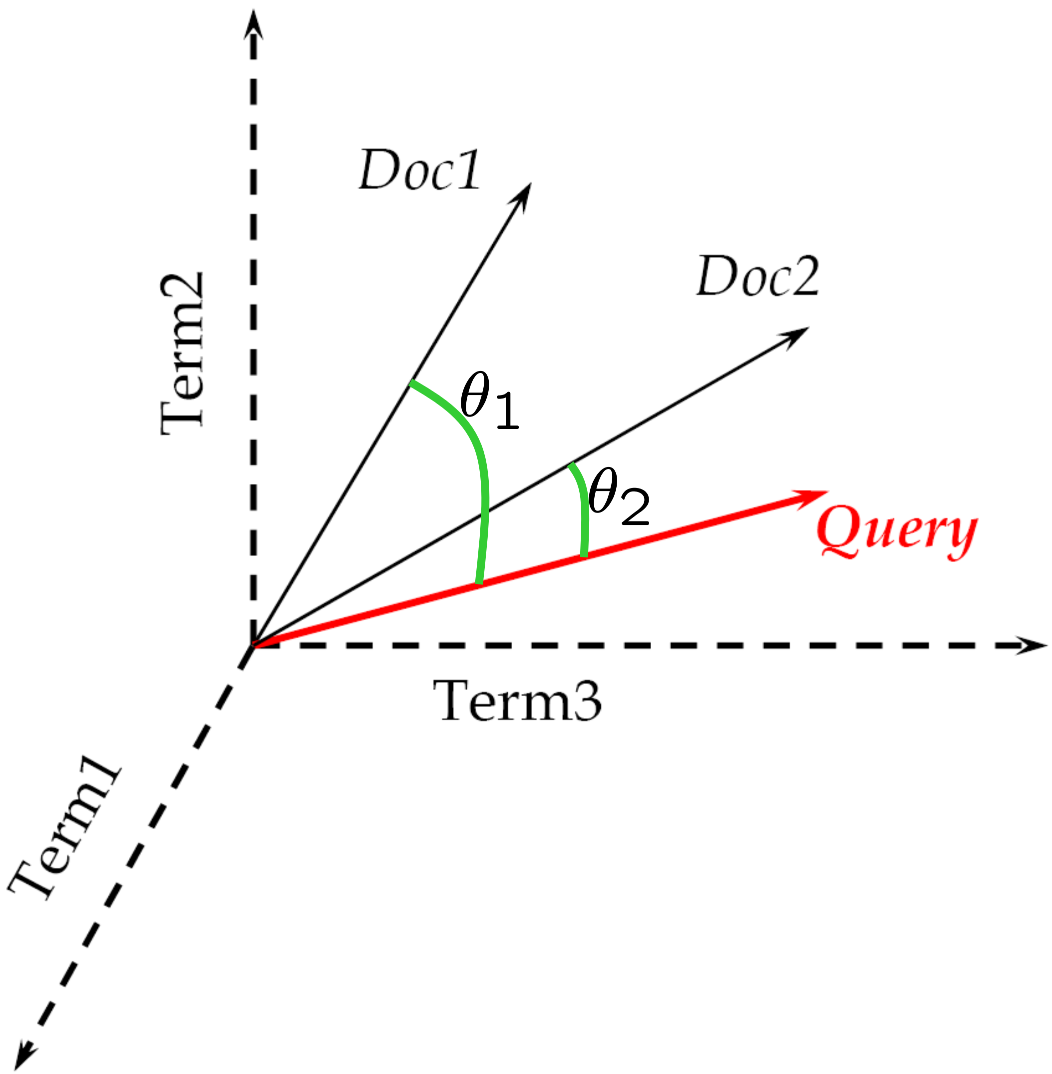
Orthogonal Terms

Term	cat	dog	lion
cat	1.00	0.00	0.00
dog	0.00	1.00	0.00
lion	0.00	0.00	1.00

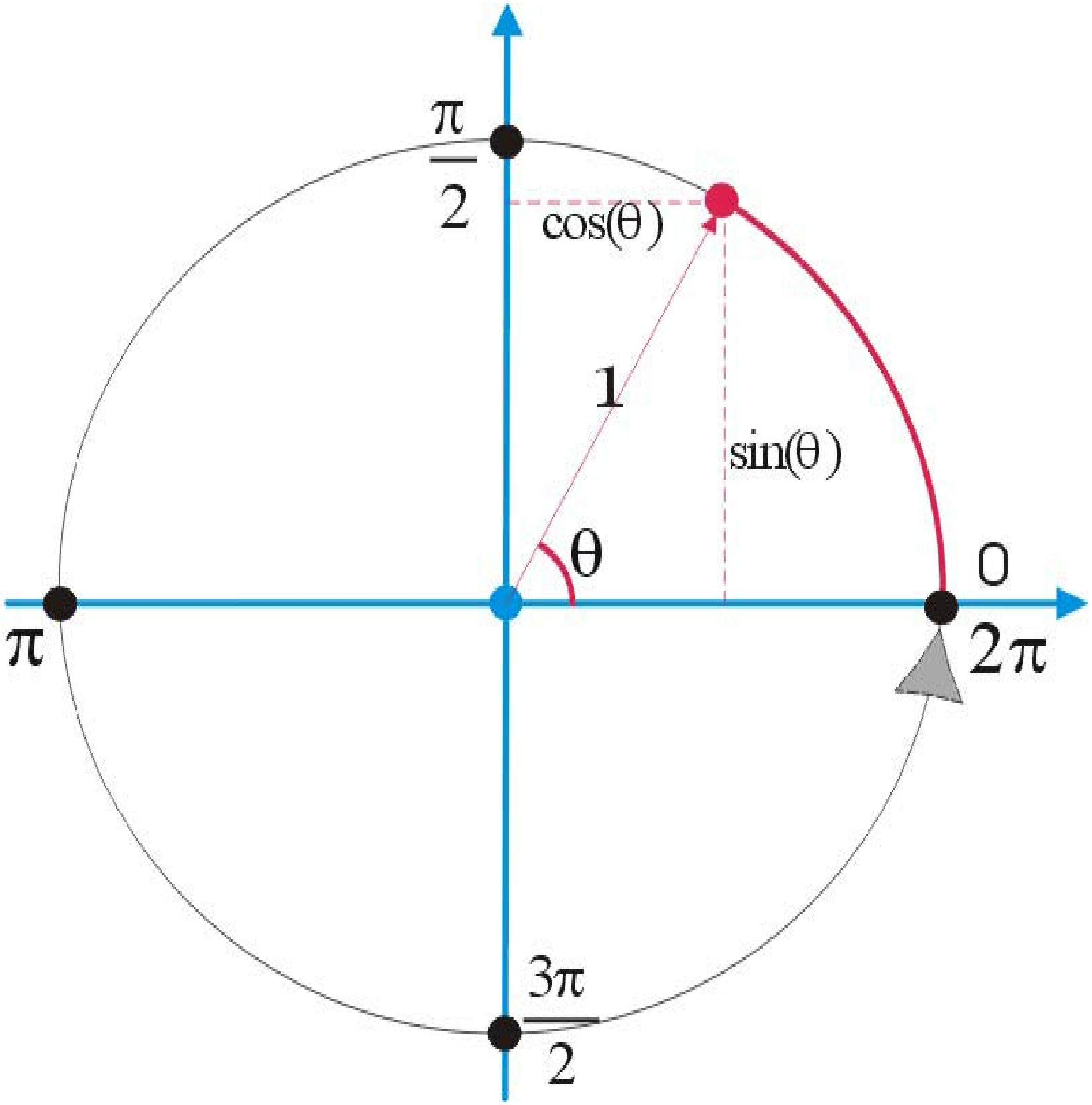
$$
\begin{aligned}
\mathrm{Sim}(D_1,Q) &= (3T_1 + 1T_2 + 4T_3) \bullet (2T_2) \\
&= 6T_1 \bullet T_2 + 2T_2 \bullet T_2 + 8T_3 \bullet T_2 \\
&= -6 \bullet 0.2 + 2 \bullet 1 \quad - \ 8 \bullet 0.4 \\
&= -1.2 \qquad + 2 \qquad - \ 3.2 \\
&= -2.4
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Sim}(D_1,Q) &= 3 \bullet 0 + 1 \bullet 2 + 4 \bullet 0 \\
&= 2
\end{aligned}
$$

vector similarity: cosine



cosine, normalization



cosine similarity: example

$D_1 = (0.5T_1 + 0.8T_2 + 0.3T_3)$ $\quad$ $Q = (1.5T_1 + 1T_2 + 0T_3)$

$$\mathrm{Sim}(D_1,Q) = \frac{(0.5\times 1.5)+(0.8\times 1)}{\sqrt{(0.5^2+0.8^2+0.3^2)(1.5^2+1^2)}}$$

$$= \frac{1.55}{\sqrt{.98\times 3.25}}$$

$$= .868$$

cosine example normalized

$$D_1 = (0.5T_1 + 0.8T_2 + 0.3T_3) \qquad Q = (1.5T_1 + 1T_2 + 0T_3)$$

$$\begin{aligned} D_1' &= (0.5T_1 + 0.8T_2 + 0.3T_3)/\sqrt{0.98} \\ &\approx 0.51T_1 + 0.82T_2 + 0.31T_3 \end{aligned} \qquad \begin{aligned} Q' &= (1.5T_1 + 1T_2 + 0T_3)/\sqrt{3.25} \\ &\approx 0.83T_1 + 0.555T_2 \end{aligned}$$

$$\begin{aligned} \text{Sim}(D_1, Q) &= \text{Sim}(D_1', Q') \\ &= \frac{(0.51 \times 0.83) + (0.82 \times 0.555)}{\sqrt{(0.51^2 + 0.82^2 + 0.31^2)(0.83^2 + 0.555^2)}} \\ &= (0.51 \times 0.83) + (0.82 \times 0.555) \\ &= 0.878 \\ &\approx 0.868 \text{ (from earlier slide)} \end{aligned}$$

round-off error,
should be the same

tf-idf base similarity formula

$$\frac{\sum_t (\mathbf{TF}_{query}(t) \cdot \mathbf{IDF}_{query}(t)) \cdot (\mathbf{TF}_{doc}(t) \cdot \mathbf{IDF}_{doc}(t))}{||doc|| \cdot ||query||}$$

- many options for TF_{query} and TF_{doc}
  - raw tf, Robertson tf, Lucene etc
  - try to come up with yours
- some options for IDF_{doc}
- $\text{IDF}_{\text{query}}$ sometimes not considered
- normalization is critical

Lucene comparison

User-specified boost

tf·idf from document

$$\sum_t \left(\frac{\mathrm{tf}_{q,t} \cdot \mathrm{idf}_t}{\mathrm{norm}_q} \cdot \frac{\mathrm{tf}_{d,t} \cdot \mathrm{idf}_t}{\mathrm{norm}_{d,t}} \cdot \mathrm{boost}_t \right) \cdot \frac{\mathrm{overlap}(q,d)}{|q|}$$

Proportion of query matched

Length-normalized query weight

Term normalization is square root of number of tokens in *d* that are in the same field as *t*

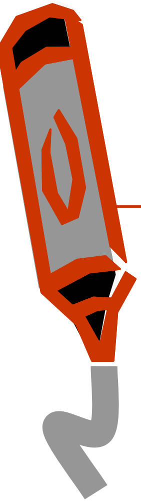
other term weighting schemes

- Lucene

$$w_{t,d} = \frac{\mathsf{tf}_{d,t} \cdot \log(N/\mathsf{df}_t + 1)}{\sqrt{\text{number of tokens in } d \text{ in the same field as } t}}$$

- augmented tf-idf cosine

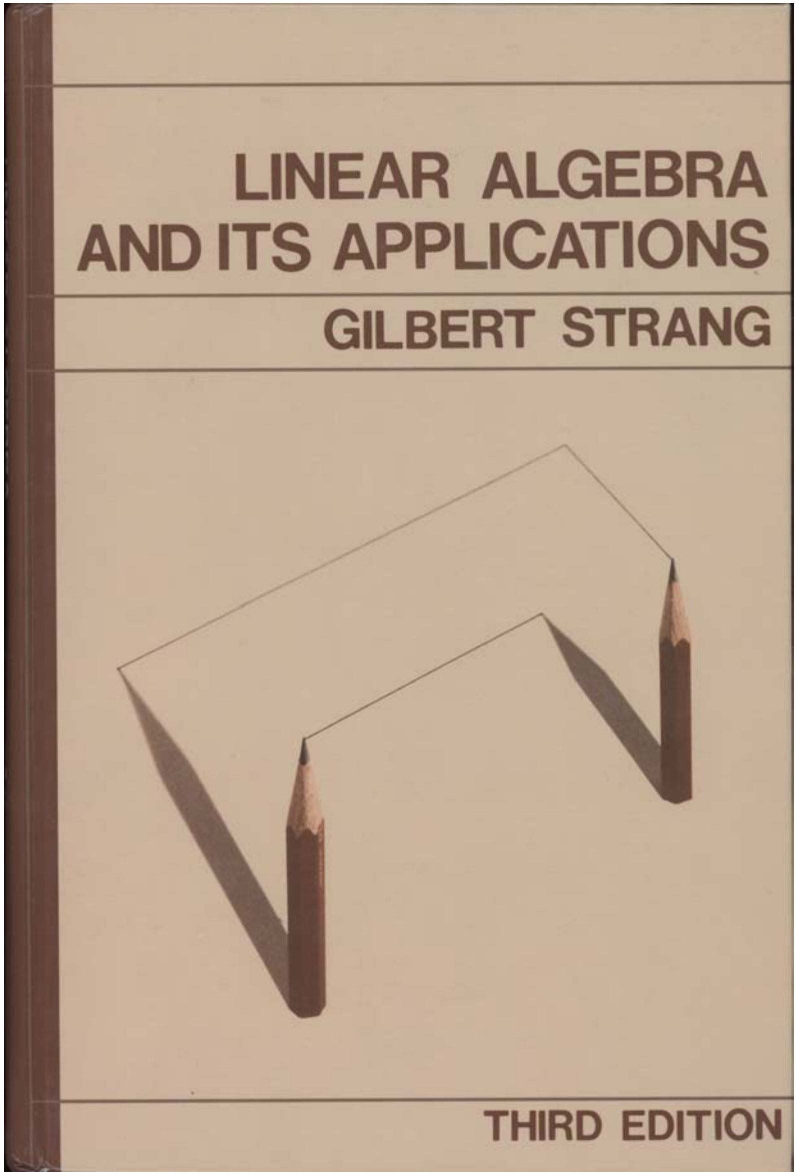
$$\frac{\left(\frac{1}{2} + \frac{1}{2}\frac{\mathsf{tf}_{t,d}}{\max(\mathsf{tf}_{*,d})}\right) \cdot \log\frac{N}{n_t}}{\left[\sum_t \left(\left(\frac{1}{2} + \frac{1}{2}\frac{\mathsf{tf}_{t,d}}{\max(\mathsf{tf}_{*,d})}\right) \cdot \log\frac{N}{n_t}\right)^2\right]^{0.5}}$$



outline

- review: geometry, linear algebra
- vector space model
- vector selection
- simmilarity
- weighting schemes
- latent semnatic indexing

more linear algebra



$A = LDU$ factorization

- For any mxn matrix A, there exists a permutation matrix P, a lower triangular matrix L with unit diagonal and an mxn echelon matrix U such that $PA = LU$

$$U = \begin{bmatrix} * & * & * & * & * & * & * & * & * \\ 0 & * & * & * & * & * & * & * & * \\ 0 & 0 & 0 & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- For any nxn matrix A, there exists L,U lower and upper triunghiular with unit diagonals, D a diagonal matrix of pivots and P a permutation matrix such that $PA = LDU$

- If A is symmetric $(A = A^T)$ then there is no need for P and $U = L^T$: $A = LDL^T$

eigenvalues and eigenvectors

- λ is an eigenvalue for matrix A iff $det(A - \lambda I) = 0$
- every eigenvalue has a correspondent non-zero eigenvector x that satisfies $(A - \lambda I)x = 0$ or $Ax = \lambda x$ in other words Ax and x have same direction
- sum of eigenvalues $=$ trace(A) $=$ sum of diagonal
- product of eigenvalues $=$ det(A)
- eigenvalues of a upper/lower triangular matrix are the diagonal entries

matrix diagonal form

- if A has lineary independent eigenvectors $y_1, y_2, ..., y_n$ and S is the matrix having those as columns, $S = [y_1 y_2 ... y_n]$, then S is invertible and
$$S^{-1}AS = \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \cdots & \\ & & & \lambda_n \end{bmatrix}$$
, the diagonal matrix of eigenvalues of A.
- $A = S\Lambda S^{-1}$
- no repeated eigenval $\Rightarrow$ indep. eigenvect
- A symetric $A^T = A \Rightarrow S$ orthogonal: $S^T S = 1$
- S is not unique
- $AS = S\Lambda$ holds iff S has eigenvect as columns
- not all matrices are diagonalizable

singular value decomposition

- if A is $m \times n, m > n$ real matrix then it can be decomposed as
$A = UDV^T$ where

- U is $m \times n$; D, V are $n \times n$
- U, V are orthogonal : $U^T U = V^T V = 1_{n \times n}$
- D is diagonal, its entries are the squre roots of eigenvalues of $A^T A$

latent semantic indexing

- Variant of the vector space model

- Uses Singular Value Decomposition (a dimensionality reduction technique) to identify uncorrelated, significant basis vectors or factors
  - Rather than non-independent terms

- Replace original words with a subset of the new factors (say 100) in both documents and queries

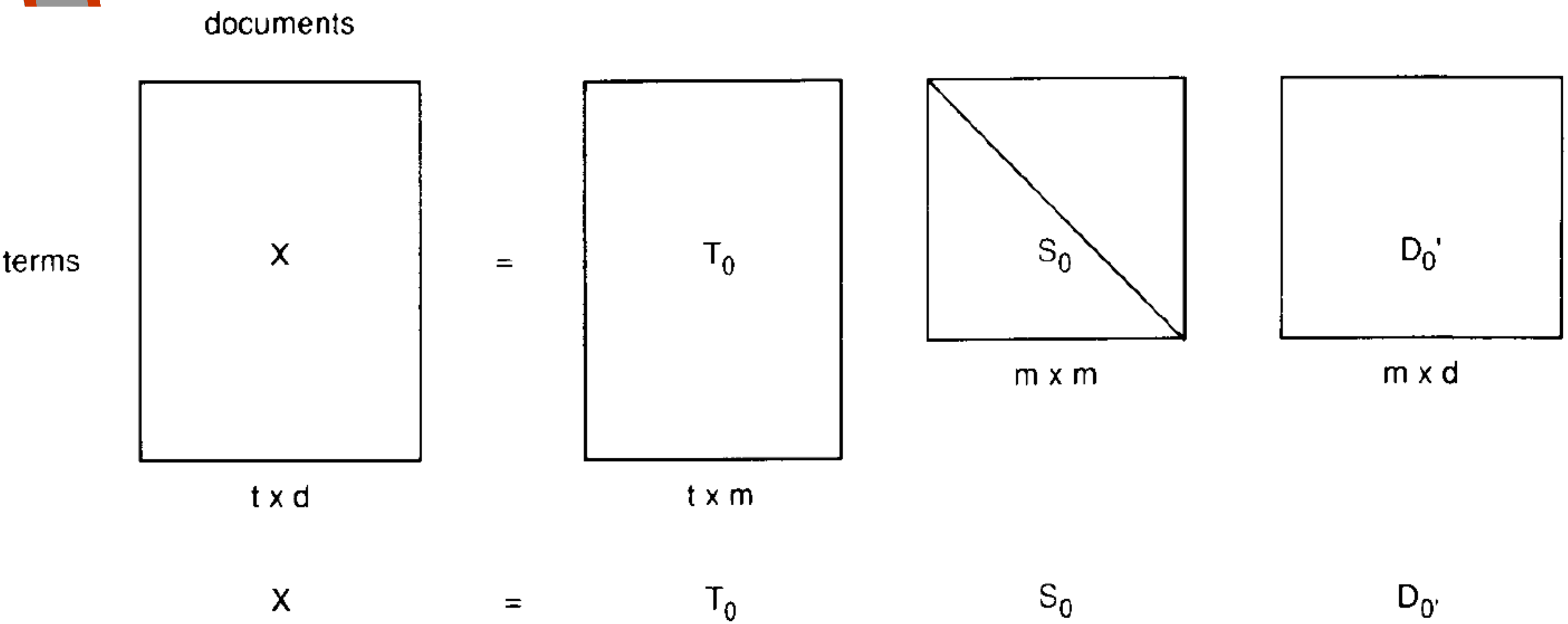
- Compute similarities in this new space

- Computationally expensive, uncertain effectiveness

dimensionality reduction

- when the representation space is rich
- but the data is lying in a small-dimension subspace
- that's when some eigenvalues are zero
- non-exact: ignore smallest eigenvalues, even if they are not zero

latent semantic indexing



- T_0,D_0 orthogonal matrices with unit length columns ($T_0 * T_0^T = 1$)
- S_0 diagonal matrix of eigen values
- m is the rank of X

LSI: example

c1:	*Human* machine *interface* for Lab 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, unordered *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*

Terms	Documents								
	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

LSI: example

$T_0 =$

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

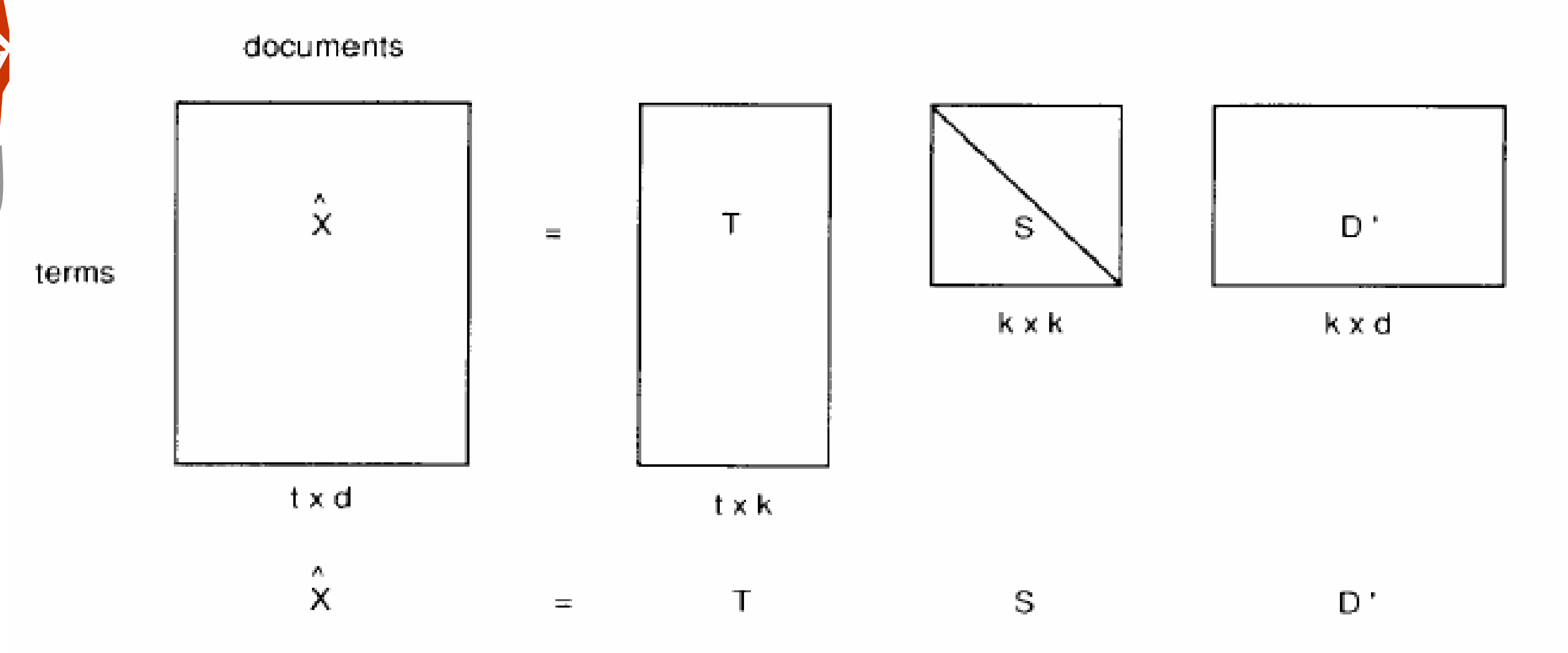
$S_0 =$

$$
\begin{matrix}
3.34 & & & & & & & & \\
 & 2.54 & & & & & & & \\
 & & 2.35 & & & & & & \\
 & & & 1.64 & & & & & \\
 & & & & 1.50 & & & & \\
 & & & & & 1.31 & & & \\
 & & & & & & 0.85 & & \\
 & & & & & & & 0.56 & \\
 & & & & & & & & 0.36
\end{matrix}
$$

$D_0 =$

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

LSI



- T has orthogonal unit-length col ($T * T^T = 1$)
- D has orthogonal unit-length col ($D * D^T = 1$)
- S diagonal matrix of eigen values
- m is the rank of X
- $t = \#$ of rows in X
- $d = \#$ of columns in X
- $k =$ chosen number of dimensions of reduced model

$$X \approx \underset{T}{\begin{matrix} 0.22 & -0.11 \\ 0.20 & -0.07 \\ 0.24 & 0.04 \\ 0.40 & 0.06 \\ 0.64 & -0.17 \\ 0.27 & 0.11 \\ 0.27 & 0.11 \\ 0.30 & -0.14 \\ 0.21 & 0.27 \\ 0.01 & 0.49 \\ 0.04 & 0.62 \\ 0.03 & 0.45 \end{matrix}} \quad \underset{S}{\begin{matrix} 3.34 & \\ & 2.54 \end{matrix}} \quad \underset{D'}{\begin{matrix} 0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.02 & 0.02 & 0.08 \\ -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \end{matrix}}$$

LSI: example

$$\hat{X} =$$

$$\begin{matrix}
0.16 & 0.40 & 0.38 & 0.47 & 0.18 & -0.05 & -0.12 & -0.16 & -0.09 \\
0.14 & 0.37 & 0.33 & 0.40 & 0.16 & -0.03 & -0.07 & -0.10 & -0.04 \\
0.15 & 0.51 & 0.36 & 0.41 & 0.24 & 0.02 & 0.06 & 0.09 & 0.12 \\
0.26 & 0.84 & 0.61 & 0.70 & 0.39 & 0.03 & 0.08 & 0.12 & 0.19 \\
0.45 & 1.23 & 1.05 & 1.27 & 0.56 & -0.07 & -0.15 & -0.21 & -0.05 \\
0.16 & 0.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\
0.16 & 0.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\
0.22 & 0.55 & 0.51 & 0.63 & 0.24 & -0.07 & -0.14 & -0.20 & -0.11 \\
0.10 & 0.53 & 0.23 & 0.21 & 0.27 & 0.14 & 0.31 & 0.44 & 0.42 \\
-0.06 & 0.23 & -0.14 & -0.27 & 0.14 & 0.24 & 0.55 & 0.77 & 0.66 \\
-0.06 & 0.34 & -0.15 & -0.30 & 0.20 & 0.31 & 0.69 & 0.98 & 0.85 \\
-0.04 & 0.25 & -0.10 & -0.21 & 0.15 & 0.22 & 0.50 & 0.71 & 0.62
\end{matrix}$$

original vs LSI

	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

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

using LSI

$X \approx$

$$\underset{T}{\begin{bmatrix} 0.22 & -0.11 \\ 0.20 & -0.07 \\ 0.24 & 0.04 \\ 0.40 & 0.06 \\ 0.64 & -0.17 \\ 0.27 & 0.11 \\ 0.27 & 0.11 \\ 0.30 & -0.14 \\ 0.21 & 0.27 \\ 0.01 & 0.49 \\ 0.04 & 0.62 \\ 0.03 & 0.45 \end{bmatrix}} \underset{S}{\begin{bmatrix} 3.34 & \\ & 2.54 \end{bmatrix}} \underset{D'}{\begin{bmatrix} 0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.02 & 0.02 & 0.08 \\ -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \end{bmatrix}}$$

- D is new doc vectors (k dimensions)
- T provides term vectors
- Given $Q=q_1 q_2 \ldots q_t$ want to compare to docs
- Convert Q from t dimensions to k

$$Q' = Q^T_{1\times t} * T_{t\times k} * S^{-1}_{k\times k}$$

- Can now compare to doc vectors
- Same basic approach can be used to add new docs to the database

LSI: does it work?

- Decomposes language into “basis vectors”
  - In a sense, is looking for core concepts

- In theory, this means that system will retrieve documents using synonyms of your query words
  - The “magic” that appeals to people

- From a demo at lsi.research.telcordia.com
  - They hold the patent on LSI

vector space retrieval: summary

- Standard vector space
  - Each dimension corresponds to a term in the vocabulary
  - Vector elements are real-valued, reflecting term importance
  - Any vector (document,query, ...) can be compared to any other
  - Cosine correlation is the similarity metric used most often

- Latent Semantic Indexing (LSI)
  - Each dimension corresponds to a "basic concept"
  - Documents and queries mapped into basic concepts
  - Same as standard vector space after that
  - Whether it's good depends on what you want

vector space model: disadvantages

- Assumed independence relationship among terms
  - Though this is a *very* common retrieval model assumption

- Lack of justification for some vector operations
  - e.g. choice of similarity function
  - e.g., choice of term weights

- Barely a retrieval model
  - Doesn’t explicitly model relevance, a person’s information need, language models, etc.

- Assumes a query and a document can be treated the same (symmetric)

vector space model: advantages

- Simplicity

- Ability to incorporate term weights
  - *Any* type of term weights can be added
  - No model that has to justify the use of a weight

- Ability to handle "distributed" term representations
  - e.g., LSI

- Can measure similarities between almost anything:
  - documents and queries
  - documents and documents
  - queries and queries
  - sentences and sentences
  - etc.