

Beyond Bag of Words III: Dimensionality Reduction for IR

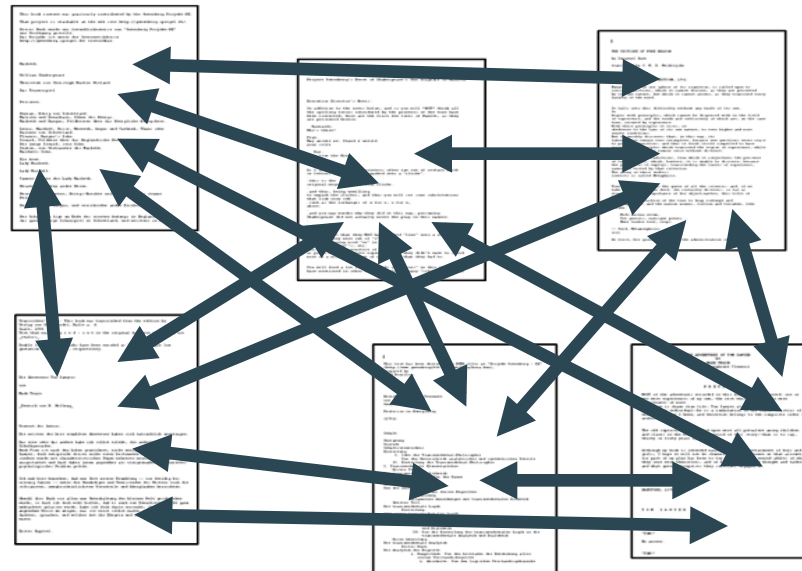
CS6200

With slides from

Kriste Krstovski, Christopher Manning, and Pandu Nayak

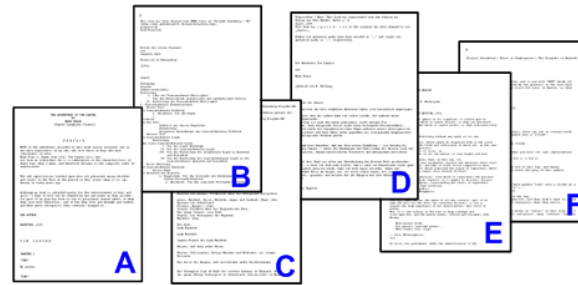
Introduction

- ▶ Document similarity – common task in:
 - ▶ Information Retrieval (IR)
 - ▶ Natural Language Processing (NLP)
 - ▶ Machine Translation (MT)



Introduction

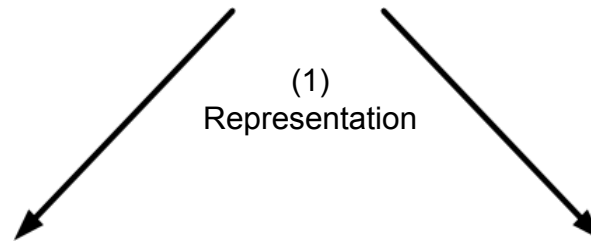
- Document similarity – how do we do it?



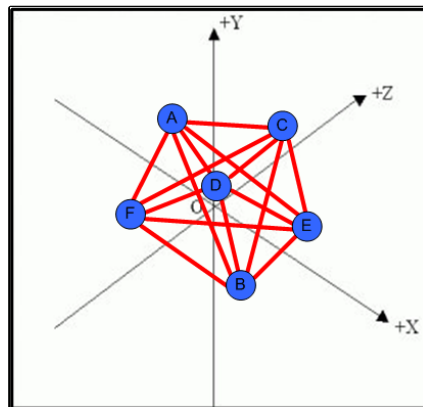
Large Document Collection

Vector Space Model
 $[tfidf_{11}, 0, 0, 0, 0, 0]$
 $[tfidf_{14}, \dots, 0, 0]$
sparse representation

Latent Dirichlet Allocation (LDA)
 $[\theta_{11}, \theta_{12}, \theta_{13}, \dots, \theta_{1n-1}, \theta_{1n}]$
dense representation

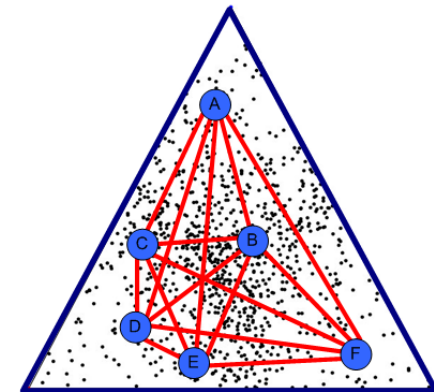


(1)
Representation



Metric Space

(2)
Comparison

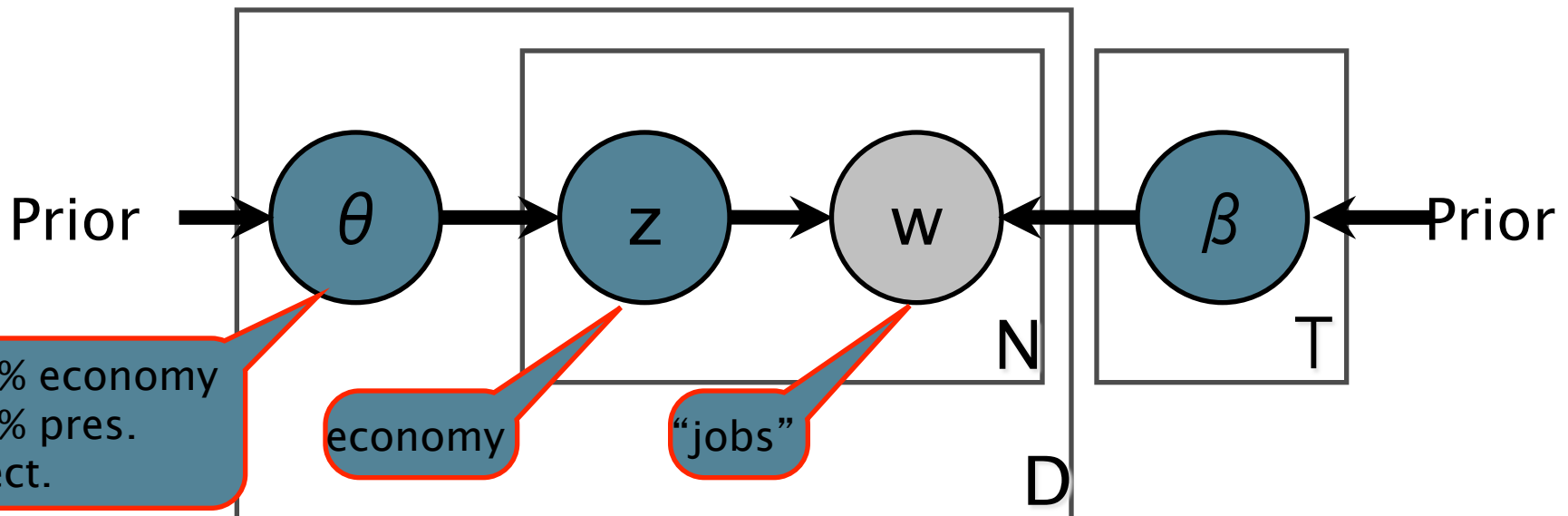


Probability Simplex

Modeling Text with Topics

Latent Dirichlet Allocation (Blei, Ng, Jordan 2003)

- Let the text talk about T topics
- Each topic is a probability distribution over all words
- For D documents each with N_D words:



Top Words by Topic

Topics →

1	2	3	4	5	6	7	8
DISEASE	WATER	MIND	STORY	FIELD	SCIENCE	BALL	JOB
BACTERIA	FISH	WORLD	STORIES	MAGNETIC	STUDY	GAME	WORK
DISEASES	SEA	DREAM	TELL	MAGNET	SCIENTISTS	TEAM	JOBS
GERMS	SWIM	DREAMS	CHARACTER	WIRE	SCIENTIFIC	FOOTBALL	CAREER
FEVER	SWIMMING	THOUGHT	CHARACTERS	NEEDLE	KNOWLEDGE	BASEBALL	EXPERIENCE
CAUSE	POOL	IMAGINATION	AUTHOR	CURRENT	WORK	PLAYERS	EMPLOYMENT
CAUSED	LIKE	MOMENT	READ	COIL	RESEARCH	PLAY	OPPORTUNITIES
SPREAD	SHELL	THOUGHTS	TOLD	POLES	CHEMISTRY	FIELD	WORKING
VIRUSES	SHARK	OWN	SETTING	IRON	TECHNOLOGY	PLAYER	TRAINING
INFECTION	TANK	REAL	TALES	COMPASS	MANY	BASKETBALL	SKILLS
VIRUS	SHELLS	LIFE	PLOT	LINES	MATHEMATICS	COACH	CAREERS
MICROORGANISMS	SHARKS	IMAGINE	TELLING	CORE	BIOLOGY	PLAYED	POSITIONS
PERSON	DIVING	SENSE	SHORT	ELECTRIC	FIELD	PLAYING	FIND
INFECTIOUS	DOLPHINS	CONSCIOUSNESS	FICTION	DIRECTION	PHYSICS	HIT	POSITION
COMMON	SWAM	STRANGE	ACTION	FORCE	LABORATORY	TENNIS	FIELD
CAUSING	LONG	FEELING	TRUE	MAGNETS	STUDIES	TEAMS	OCCUPATIONS
SMALLPOX	SEAL	WHOLE	EVENTS	BE	WORLD	GAMES	REQUIRE
BODY	DIVE	BEING	TELLS	MAGNETISM	SCIENTIST	SPORTS	OPPORTUNITY
INFECTIONS	DOLPHIN	MIGHT	TALE	POLE	STUDYING	BAT	EARN
CERTAIN	UNDERWATER	HOPE	NOVEL	INDUCED	SCIENCES	TERRY	ABLE

Griffiths et al.

Top Words by Topic

Topics →

1	2	3	4	5	6	7	8
DISEASE	WATER	MIND	STORY	FIELD	SCIENCE	BALL	JOB
BACTERIA	FISH	WORLD	STORIES	MAGNETIC	STUDY	GAME	WORK
DISEASES	SEA	DREAM	TELL	MAGNET	SCIENTISTS	TEAM	JOBS
GERMS	SWIM	DREAMS	CHARACTER	WIRE	SCIENTIFIC	FOOTBALL	CAREER
FEVER	SWIMMING	THOUGHT	CHARACTERS	NEEDLE	KNOWLEDGE	BASEBALL	EXPERIENCE
CAUSE	POOL	IMAGINATION	AUTHOR	CURRENT	WORK	PLAYERS	EMPLOYMENT
CAUSED	LIKE	MOMENT	READ	COIL	RESEARCH	PLAY	OPPORTUNITIES
SPREAD	SHELL	THOUGHTS	TOLD	POLES	CHEMISTRY	FIELD	WORKING
VIRUSES	SHARK	OWN	SETTING	IRON	TECHNOLOGY	PLAYER	TRAINING
INFECTION	TANK	REAL	TALES	COMPASS	MANY	BASKETBALL	SKILLS
VIRUS	SHELLS	LIFE	PLOT	LINES	MATHEMATICS	COACH	CAREERS
MICROORGANISMS	SHARKS	IMAGINE	TELLING	CORE	BIOLOGY	PLAYED	POSITIONS
PERSON	DIVING	SENSE	SHORT	ELECTRIC	FIELD	PLAYING	FIND
INFECTIOUS	DOLPHINS	CONSCIOUSNESS	FICTION	DIRECTION	PHYSICS	HIT	POSITION
COMMON	SWAM	STRANGE	ACTION	FORCE	LABORATORY	TENNIS	FIELD
CAUSING	LONG	FEELING	TRUE	MAGNETS	STUDIES	TEAMS	OCCUPATIONS
SMALLPOX	SEAL	WHOLE	EVENTS	BE	WORLD	GAMES	REQUIRE
BODY	DIVE	BEING	TELLS	MAGNETISM	SCIENTIST	SPORTS	OPPORTUNITY
INFECTIONS	DOLPHIN	MIGHT	TALE	POLE	STUDYING	BAT	EARN
CERTAIN	UNDERWATER	HOPE	NOVEL	INDUCED	SCIENCES	TERRY	ABLE

Griffiths et al.

Today's topic

Latent Semantic Indexing

- Term-document matrices are very large
- But the number of topics that people talk about is small (in some sense)
 - Money, movies, politics, homework, ...
- Can we represent the term-document space by a lower dimensional latent space?

Linear Algebra Background

Eigenvalues & Eigenvectors

- **Eigenvectors** (for a square $m \times m$ matrix \mathbf{S})

$$\mathbf{S}\mathbf{v} = \lambda\mathbf{v}$$

(right) eigenvector
eigenvalue

$\mathbf{v} \in \mathbb{R}^m \neq \mathbf{0}$
 $\lambda \in \mathbb{R}$

Example

$$\begin{pmatrix} 6 & -2 \\ 4 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

- **How many eigenvalues** are there at most?

$$\mathbf{S}\mathbf{v} = \lambda\mathbf{v} \iff (\mathbf{S} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$$

only has a non-zero solution if $|\mathbf{S} - \lambda\mathbf{I}| = 0$

This is a m th order equation in λ which can have **at most m distinct solutions** (roots of the characteristic polynomial) - can be complex even though \mathbf{S} is real.

Matrix-vector multiplication

$S = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ has eigenvalues 30, 20, 1 with corresponding eigenvectors

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad v_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

On each eigenvector, S acts as a multiple of the identity matrix: but as a different multiple on each.

Any vector (say $x = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}$) can be viewed as a combination of the eigenvectors: $x = 2v_1 + 4v_2 + 6v_3$

Matrix-vector multiplication

- Thus a matrix-vector multiplication such as Sx (S , x as in the previous slide) can be rewritten in terms of the eigenvalues/vectors:

$$Sx = S(2v_1 + 4v_2 + 6v_3)$$

$$Sx = 2Sv_1 + 4Sv_2 + 6Sv_3 = 2\lambda_1v_1 + 4\lambda_2v_2 + 6\lambda_3v_3$$

$$Sx = 60v_1 + 80v_2 + 6v_3$$

- Even though x is an arbitrary vector, the action of S on x is determined by the eigenvalues/vectors.

Matrix-vector multiplication

- Suggestion: the effect of “small” eigenvalues is small.
- If we ignored the smallest eigenvalue (1), then instead of

$$\begin{pmatrix} 60 \\ 80 \\ 6 \end{pmatrix}$$

we would get

$$\begin{pmatrix} 60 \\ 80 \\ 0 \end{pmatrix}$$

- These vectors are similar (in cosine similarity, etc.)

Eigenvalues & Eigenvectors

For symmetric matrices, eigenvectors for distinct eigenvalues are **orthogonal**

$$Sv_{\{1,2\}} = \lambda_{\{1,2\}}v_{\{1,2\}}, \text{ and } \lambda_1 \neq \lambda_2 \Rightarrow v_1 \bullet v_2 = 0$$

All eigenvalues of a real symmetric matrix are **real**.

for complex λ , if $|S - \lambda I| = 0$ and $S = S^T \Rightarrow \lambda \in \Re$

All eigenvalues of a **positive semidefinite** matrix are **non-negative**

$$\forall w \in \Re^n, w^T S w \geq 0, \text{ then if } S v = \lambda v \Rightarrow \lambda \geq 0$$

Example

- Let $S = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ ← Real, symmetric.

- Then $S - \lambda I = \begin{bmatrix} 2 - \lambda & 1 \\ 1 & 2 - \lambda \end{bmatrix} \Rightarrow$

$$|S - \lambda I| = (2 - \lambda)^2 - 1 = 0.$$

- The eigenvalues are 1 and 3 (nonnegative, real).
- The eigenvectors are orthogonal (and real):

$$\begin{pmatrix} 1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Plug in these values and solve for eigenvectors.

Eigen/diagonal Decomposition

- Let $S \in \mathbb{R}^{m \times m}$ be a **square** matrix with **m linearly independent eigenvectors** (a “non-defective” matrix)

- Theorem:** Exists an **eigen decomposition** *diagonal*

$$S = U\Lambda U^{-1}$$

- (cf. matrix diagonalization theorem)

- Columns of U are the **eigenvectors** of S
- Diagonal elements of Λ are **eigenvalues** of S

$$\Lambda = \text{diag}(\lambda_1, \dots, \lambda_m), \quad \lambda_i \geq \lambda_{i+1}$$

Unique
for
distinct
eigen-
values

Diagonal decomposition: why/how

Let \mathbf{U} have the eigenvectors as columns:

$$\mathbf{U} = \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_n \end{bmatrix}$$

Then, \mathbf{SU} can be written

$$\mathbf{SU} = \mathbf{S} \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \lambda_1 \mathbf{v}_1 & \dots & \lambda_n \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \dots & \\ & & \lambda_n \end{bmatrix}$$

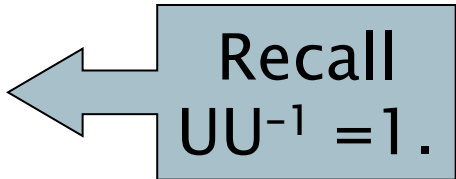
Thus $\mathbf{SU} = \mathbf{U}\mathbf{\Lambda}$, or $\mathbf{U}^{-1}\mathbf{SU} = \mathbf{\Lambda}$

And $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$.

Diagonal decomposition - example

Recall $S = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}; \lambda_1 = 1, \lambda_2 = 3.$

The eigenvectors $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ form $U = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$

Inverting, we have $U^{-1} = \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \end{bmatrix}$ 

Then, $S = U\Lambda U^{-1} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \end{bmatrix}$

Example continued

Let's divide \mathbf{U} (and multiply \mathbf{U}^{-1}) by $\sqrt{2}$

$$\text{Then, } \mathbf{S} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

$\mathbf{Q} \qquad \qquad \Lambda \qquad \qquad (\mathbf{Q}^{-1} = \mathbf{Q}^T)$

Why? Stay tuned ...

Symmetric Eigen Decomposition

- If $S \in \mathbb{R}^{m \times m}$ is a **symmetric** matrix:
- **Theorem**: There exists a (unique) **eigen decomposition** $S = Q\Lambda Q^T$
- where Q is **orthogonal**:
 - $Q^{-1} = Q^T$
 - Columns of Q are normalized eigenvectors
 - Columns are orthogonal.
 - (everything is real)

Exercise

- Examine the symmetric eigen decomposition, if any, for each of the following matrices:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 \\ -2 & 3 \end{bmatrix} \quad \begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix}$$

Time out!

- I came to this class to learn about text retrieval and mining, not to have my linear algebra past dredged up again ...
 - But if you want to dredge, Strang's *Applied Mathematics* is a good place to start.
- What do these matrices have to do with text?
- Recall $M \times N$ term-document matrices ...
- But everything so far needs square matrices – so ...

Similarity \rightarrow Clustering

- We can compute the similarity between two document vector representations x_i and x_j by $x_i x_j^T$
- Let $X = [x_1 \dots x_N]$
- Then XX^T is a matrix of similarities
- XX^T is symmetric
- So $XX^T = Q\Lambda Q^T$
- So we can decompose this similarity space into a set of orthonormal basis vectors (given in Q) scaled by the eigenvalues in Λ
 - This leads to PCA (Principal Components Analysis)

Singular Value Decomposition

For an $M \times N$ matrix \mathbf{A} of rank r there exists a factorization (Singular Value Decomposition = **SVD**) as follows:

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

The diagram illustrates the dimensions of the matrices in the SVD equation $A = U \Sigma V^T$. Three boxes are positioned below the equation, each containing a dimension: $M \times M$, $M \times N$, and $V \text{ is } N \times N$. Arrows point from the first box to the matrix U , from the second box to the matrix Σ , and from the third box to the matrix V^T .

(Not proven here.)

Singular Value Decomposition

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

$M \times M$

$M \times N$

$V \text{ is } N \times N$

- $AA^T = Q \Lambda Q^T$
- $AA^T = (U \Sigma V^T)(U \Sigma V^T)^T = (U \Sigma V^T)(V \Sigma U^T) = U \Sigma^2 U^T$

The columns of \mathbf{U} are orthogonal eigenvectors of \mathbf{AA}^T .

The columns of \mathbf{V} are orthogonal eigenvectors of $\mathbf{A}^T \mathbf{A}$.

Eigenvalues $\lambda_1 \dots \lambda_r$ of \mathbf{AA}^T are the eigenvalues of $\mathbf{A}^T \mathbf{A}$.

$$\sigma_i = \sqrt{\lambda_i}$$

$$\Sigma = \text{diag}(\sigma_1 \dots \sigma_r) \leftarrow \text{Singular values}$$

Singular Value Decomposition

- Illustration of SVD dimensions and sparseness

$$\underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_A = \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_U \underbrace{\begin{bmatrix} \bullet & & & & \\ & \bullet & & & \\ & & \bullet & & \\ & & & \bullet & \\ & & & & \bullet \end{bmatrix}}_\Sigma \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{V^T}$$

The diagram illustrates the SVD decomposition of a 5x3 matrix A . The matrix A is shown as a 5x3 grid of asterisks. It is equal to the product of three matrices: U (5x5), Σ (5x5), and V^T (3x3). The U matrix has a yellow shaded 5x2 submatrix in its right half. The Σ matrix has a yellow shaded 5x2 submatrix in its bottom half. The V^T matrix is a 3x3 grid of asterisks.

$$\underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_A = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_U \underbrace{\begin{bmatrix} \bullet & & & & \\ & \bullet & & & \\ & & \bullet & & \\ & & & \bullet & \\ & & & & \bullet \end{bmatrix}}_\Sigma \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{V^T}$$

The diagram illustrates the SVD decomposition of a 3x5 matrix A . The matrix A is shown as a 3x5 grid of asterisks. It is equal to the product of three matrices: U (3x3), Σ (3x3), and V^T (5x5). The U matrix is a 3x3 grid of asterisks. The Σ matrix has a yellow shaded 3x2 submatrix in its right half. The V^T matrix has a yellow shaded 3x5 submatrix in its bottom half.

SVD example

$$\text{Let } A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Thus $M=3$, $N=2$. Its SVD is

$$\begin{bmatrix} 0 & 2/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & 1/\sqrt{6} & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

Typically, the singular values arranged in decreasing order.

Low-rank Approximation

- SVD can be used to compute optimal **low-rank approximations**.
- Approximation problem: Find \mathbf{A}_k of rank k such that

$$\mathbf{A}_k = \min_{X: \text{rank}(X)=k} \|\mathbf{A} - X\|_F \longleftarrow \text{Frobenius norm}$$

$$\|\mathbf{A}\|_F \equiv \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}.$$

\mathbf{A}_k and X are both $m \times n$ matrices.

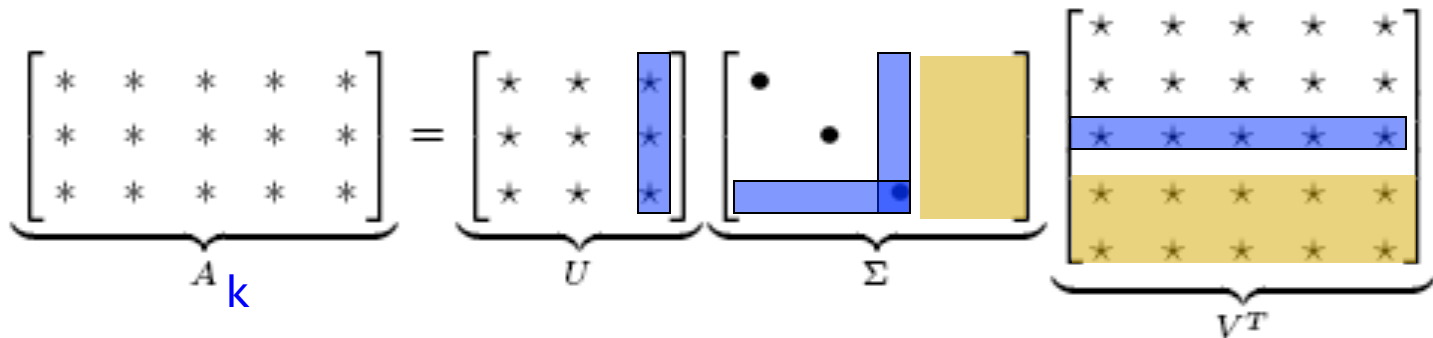
Typically, want $k \ll r$.

Low-rank Approximation

- Solution via SVD

$$A_k = U \operatorname{diag}(\sigma_1, \dots, \sigma_k, \underbrace{0, \dots, 0}_{\text{set smallest } r-k \text{ singular values to zero}}) V^T$$

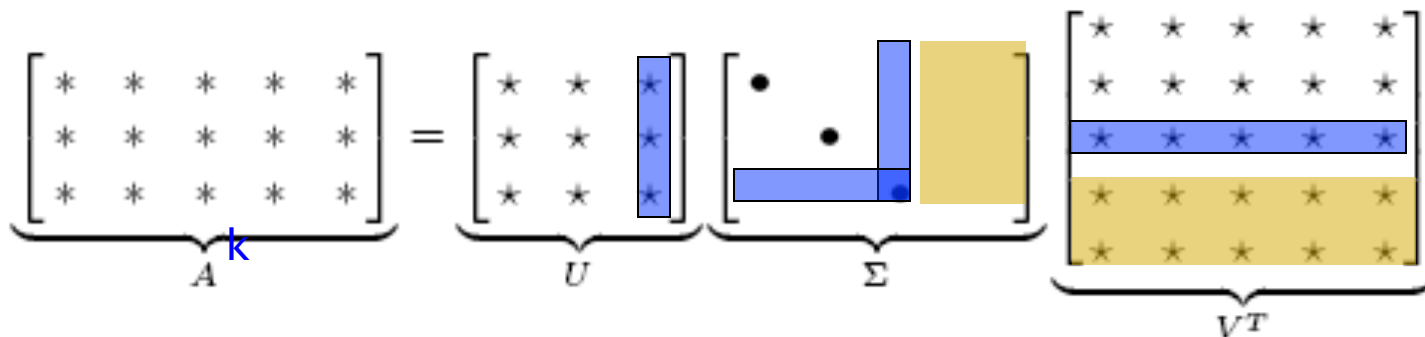
*set smallest $r-k$
singular values to zero*



$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T \leftarrow \text{column notation: sum of rank 1 matrices}$$

Reduced SVD

- If we retain only k singular values, and set the rest to 0, then we don't need the matrix parts in color
- Then Σ is $k \times k$, U is $M \times k$, V^T is $k \times N$, and A_k is $M \times N$
- This is referred to as the reduced SVD
- It is the convenient (space-saving) and usual form for computational applications
- It's what Matlab gives you



Approximation error

- How good (bad) is this approximation?
- It's the best possible, measured by the Frobenius norm of the error:

$$\min_{X:\text{rank}(X)=k} \|A - X\|_F = \|A - A_k\|_F = \sigma_{k+1}$$

where the σ_i are ordered such that $\sigma_i \geq \sigma_{i+1}$.

Suggests why Frobenius error drops as k increases.

SVD Low-rank approximation

- Whereas the term-doc matrix A may have $M=50000$, $N=10$ million (and rank close to 50000)
- We can construct an approximation A_{100} with rank 100.
 - Of all rank 100 matrices, it would have the lowest Frobenius error.
- Great ... but why would we??
- Answer: *Latent Semantic Indexing*

Latent Semantic Indexing via the SVD

What it is

- From term-doc matrix A , we compute the approximation A_k .
- There is a row for each term and a column for each doc in A_k
- Thus docs live in a space of $k \ll r$ dimensions
 - These dimensions are not the original axes
- But why?

Vector Space Model: Pros

- **Automatic** selection of index terms
- **Partial matching** of queries and documents (*dealing with the case where no document contains all search terms*)
- **Ranking** according to **similarity score** (*dealing with large result sets*)
- **Term weighting** schemes (*improves retrieval performance*)
- Various extensions
 - Document clustering
 - Relevance feedback (modifying query vector)
- Geometric foundation

Problems with Lexical Semantics



The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

- Ambiguity and association in natural language
 - **Polysemy**: Words often have a **multitude of meanings** and different types of usage (*more severe in very heterogeneous collections*).
 - The vector space model is unable to discriminate between different meanings of the same word.

$$\text{sim}_{\text{true}}(d, q) < \cos(\angle(\vec{d}, \vec{q}))$$

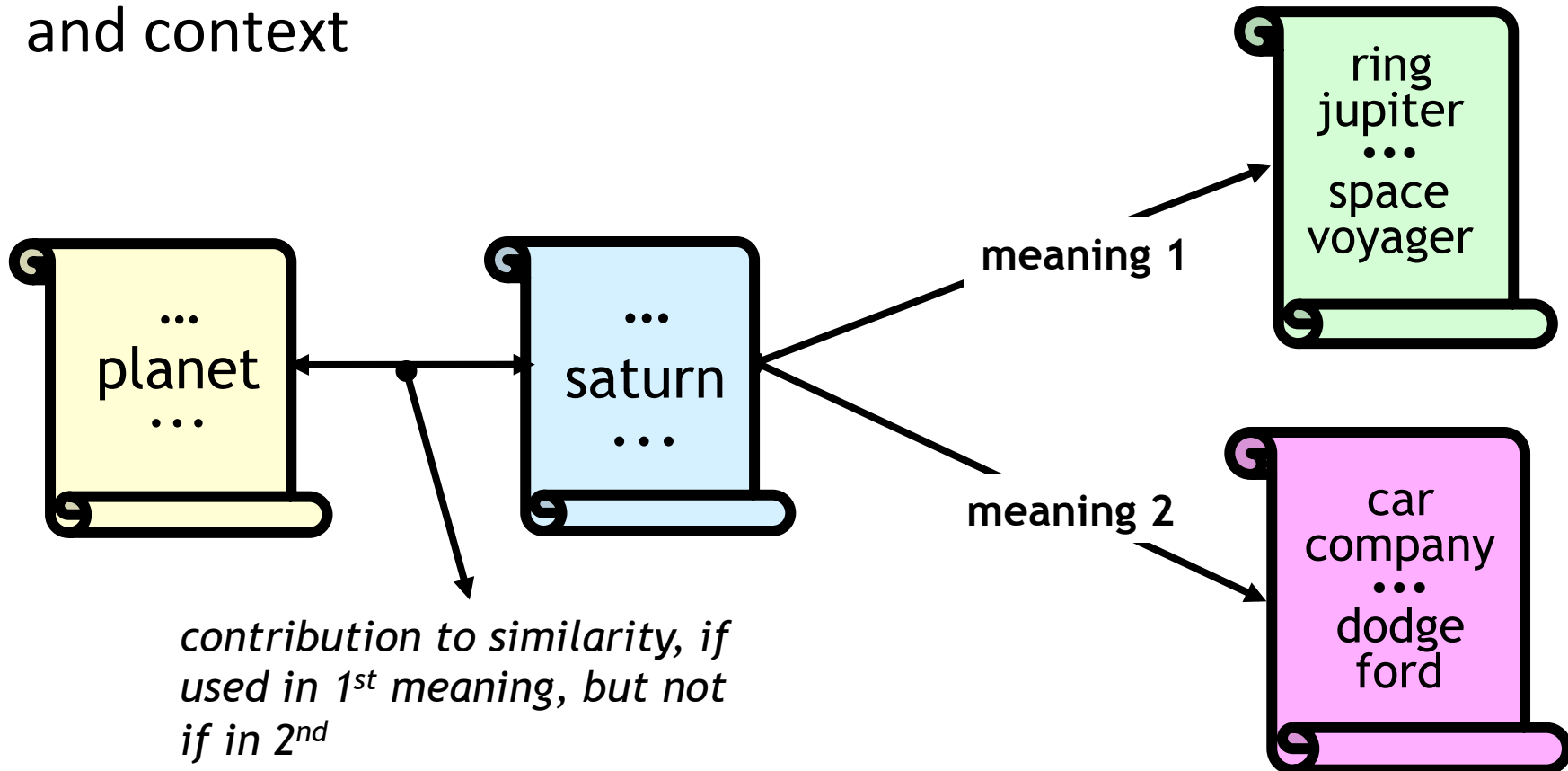
Problems with Lexical Semantics

- **Synonymy**: Different terms may have an **identical or a similar meaning** (weaker: words indicating the same topic).
- No associations between words are made in the vector space representation.

$$\text{sim}_{\text{true}}(d, q) > \cos(\angle(\vec{d}, \vec{q}))$$

Polysemy and Context

- Document similarity on single word level: polysemy and context



Latent Semantic Indexing (LSI)

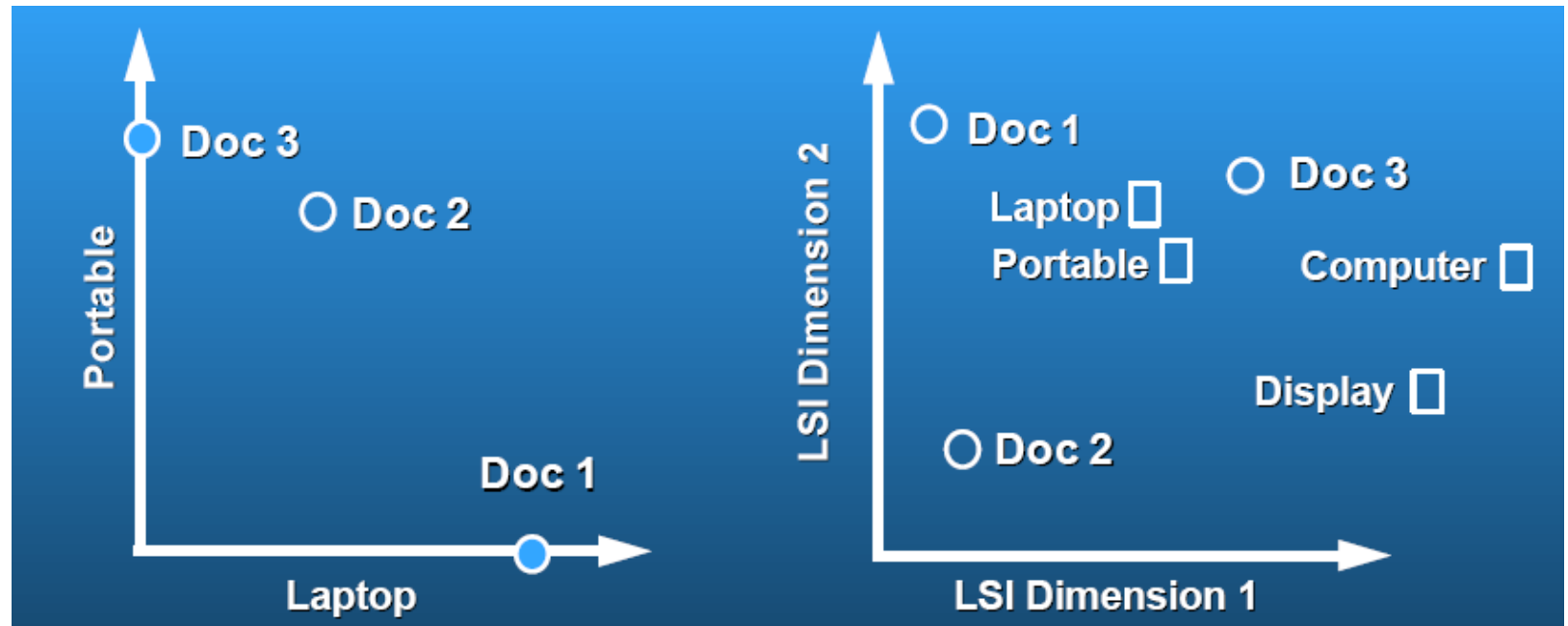
- Perform a **low-rank approximation** of **document-term matrix** (typical rank **100–300**)
- General idea
 - Map documents (*and* terms) to a **low-dimensional representation**.
 - Design a mapping such that the low-dimensional space reflects **semantic associations** (latent semantic space).
 - Compute document similarity based on the **inner product** in this **latent semantic space**

Goals of LSI

- LSI takes documents that are semantically similar (= talk about the same topics), but are not similar in the vector space (because they use different words) and re-represents them in a reduced vector space in which they have higher similarity.
- Similar terms map to similar location in low dimensional space
- Noise reduction by dimension reduction

Latent Semantic Analysis

- **Latent semantic space:** illustrating example



courtesy of Susan Dumais

Performing the maps

- Each row and column of A gets mapped into the k -dimensional LSI space, by the SVD.
- Claim – this is not only the mapping with the best (Frobenius error) approximation to A , but in fact *improves* retrieval.
- A query q is also mapped into this space, by

$$q_k = q^T U_k \Sigma_k^{-1}$$

- Query NOT a sparse vector.

LSA Example

- A simple example term-document matrix (binary)

<i>C</i>	<i>d</i> ₁	<i>d</i> ₂	<i>d</i> ₃	<i>d</i> ₄	<i>d</i> ₅	<i>d</i> ₆
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

LSA Example

- Example of $C = U\Sigma V^T$: The matrix U

U	1	2	3	4	5
ship	-0.44	-0.30	0.57	0.58	0.25
boat	-0.13	-0.33	-0.59	0.00	0.73
ocean	-0.48	-0.51	-0.37	0.00	-0.61
wood	-0.70	0.35	0.15	-0.58	0.16
tree	-0.26	0.65	-0.41	0.58	-0.09

LSA Example

- Example of $C = U\Sigma V^T$: The matrix Σ

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

LSA Example

- Example of $C = U\Sigma V^T$: The matrix V^T

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	-0.75	-0.28	-0.20	-0.45	-0.33	-0.12
2	-0.29	-0.53	-0.19	0.63	0.22	0.41
3	0.28	-0.75	0.45	-0.20	0.12	-0.33
4	0.00	0.00	0.58	0.00	-0.58	0.58
5	-0.53	0.29	0.63	0.19	0.41	-0.22

LSA Example: Reducing the dimension

U	1	2	3	4	5
ship	-0.44	-0.30	0.00	0.00	0.00
boat	-0.13	-0.33	0.00	0.00	0.00
ocean	-0.48	-0.51	0.00	0.00	0.00
wood	-0.70	0.35	0.00	0.00	0.00
tree	-0.26	0.65	0.00	0.00	0.00

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	-0.75	-0.28	-0.20	-0.45	-0.33	-0.12
2	-0.29	-0.53	-0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

Original matrix C vs. reduced $C_2 = U\Sigma_2V^T$

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	-0.08
boat	0.36	0.36	0.16	-0.20	-0.02	-0.18
ocean	1.01	0.72	0.36	-0.04	0.16	-0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	-0.39	-0.08	0.90	0.41	0.49

Why the reduced dimension matrix is better

- Similarity of d2 and d3 in the original space: 0.
- Similarity of d2 and d3 in the reduced space: $0.52 * 0.28 + 0.36 * 0.16 + 0.72 * 0.36 + 0.12 * 0.20 + -0.39 * -0.08 \approx 0.52$
- Typically, LSA increases recall and hurts precision

Empirical evidence

- Experiments on TREC 1/2/3 – Dumais
- Lanczos SVD code (available on netlib) due to Berry used in these experiments
 - Running times of \sim one day on tens of thousands of docs [still an obstacle to use!]
- Dimensions – various values 250-350 reported. Reducing k improves recall.
 - (Under 200 reported unsatisfactory)
- Generally expect recall to improve – what about precision?

Empirical evidence

- Precision at or above median TREC precision
 - Top scorer on almost 20% of TREC topics
- Slightly better on average than straight vector spaces
- Effect of dimensionality:

Dimensions	Precision
250	0.367
300	0.371
346	0.374

Failure modes

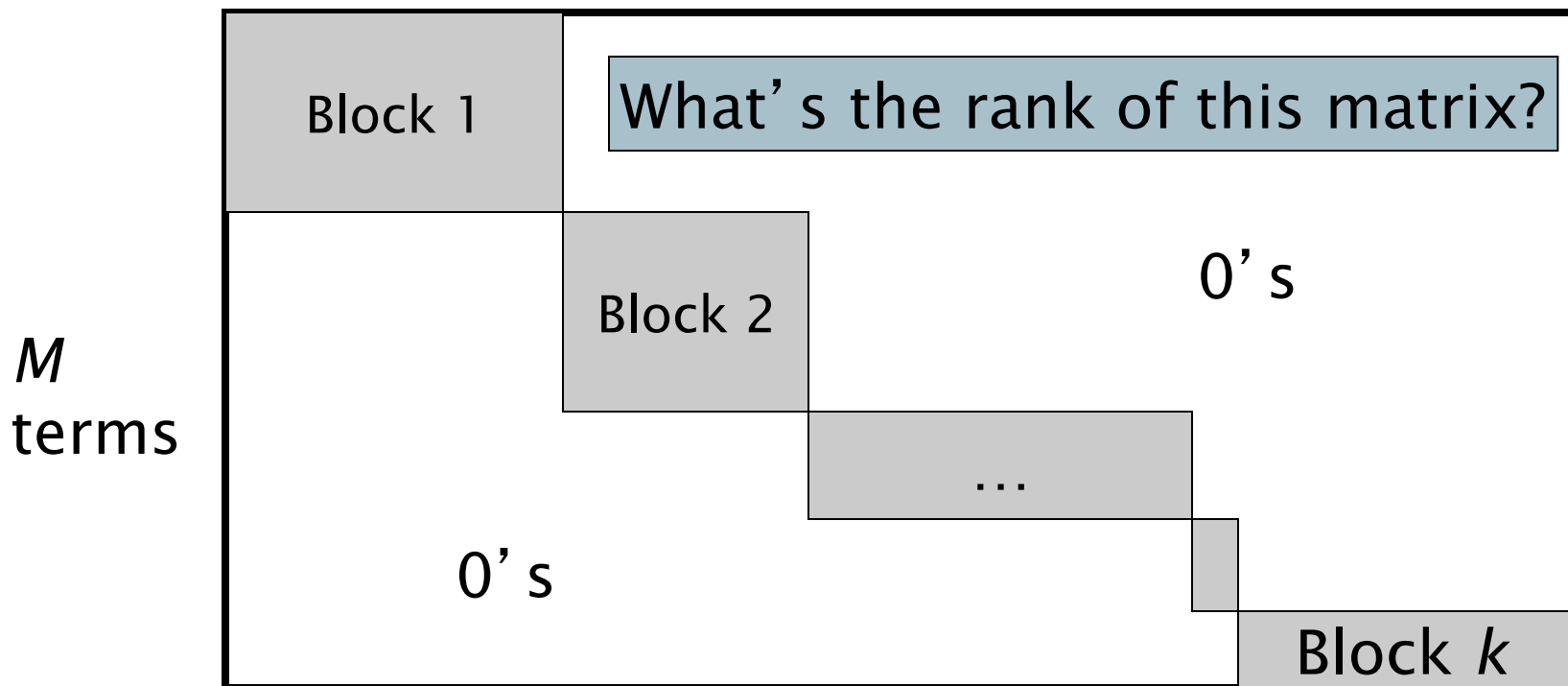
- Negated phrases
 - TREC topics sometimes negate certain query/terms phrases – precludes simple automatic conversion of topics to latent semantic space.
- Boolean queries
 - As usual, freetext/vector space syntax of LSI queries precludes (say) “Find any doc having to do with the following 5 companies”
- See Dumais for more.

But why is this clustering?

- We've talked about docs, queries, retrieval and precision here.
- What does this have to do with clustering?
- Intuition: Dimension reduction through LSI brings together “related” axes in the vector space.

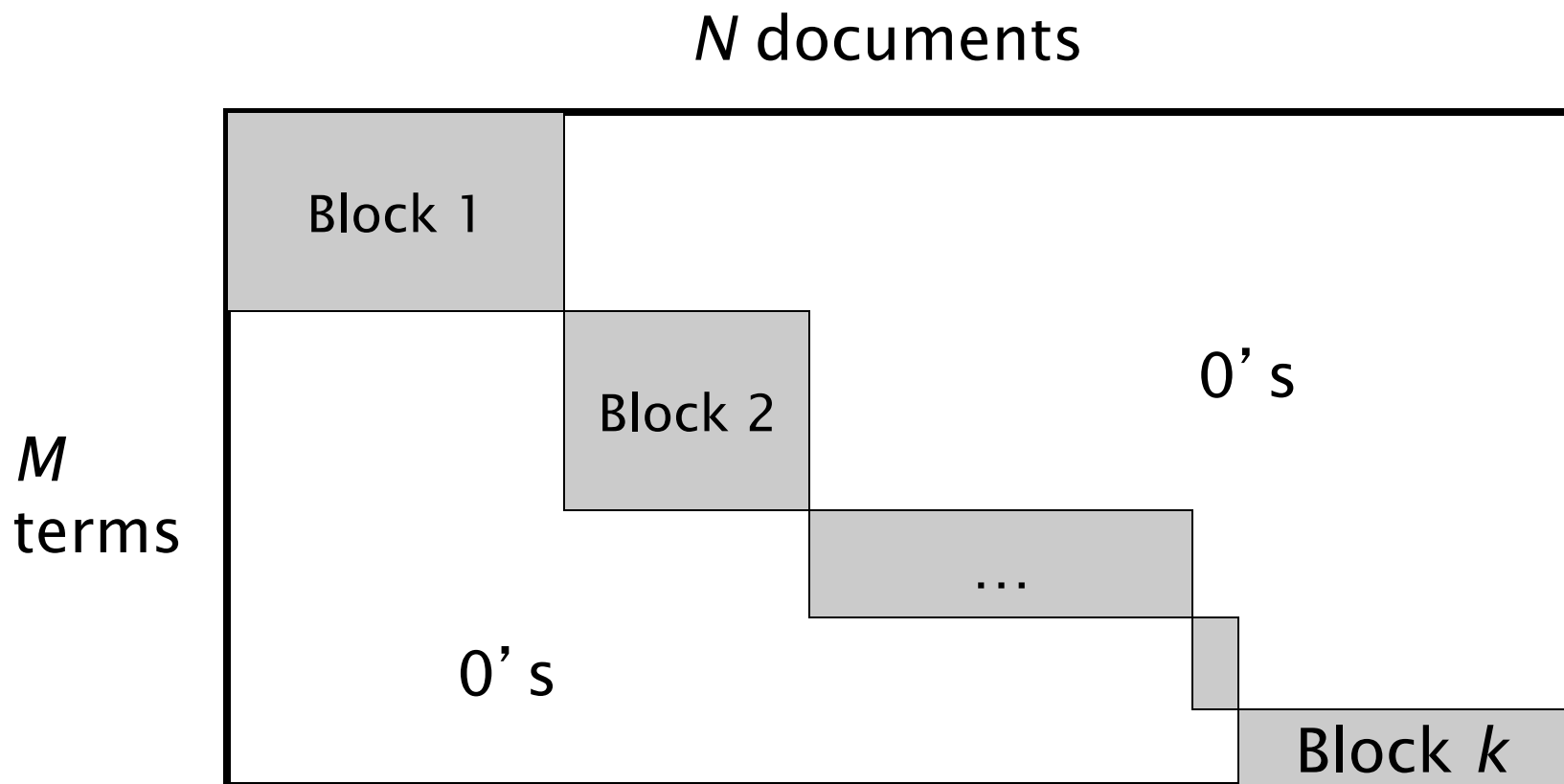
Intuition from block matrices

N documents



 = Homogeneous non-zero blocks.

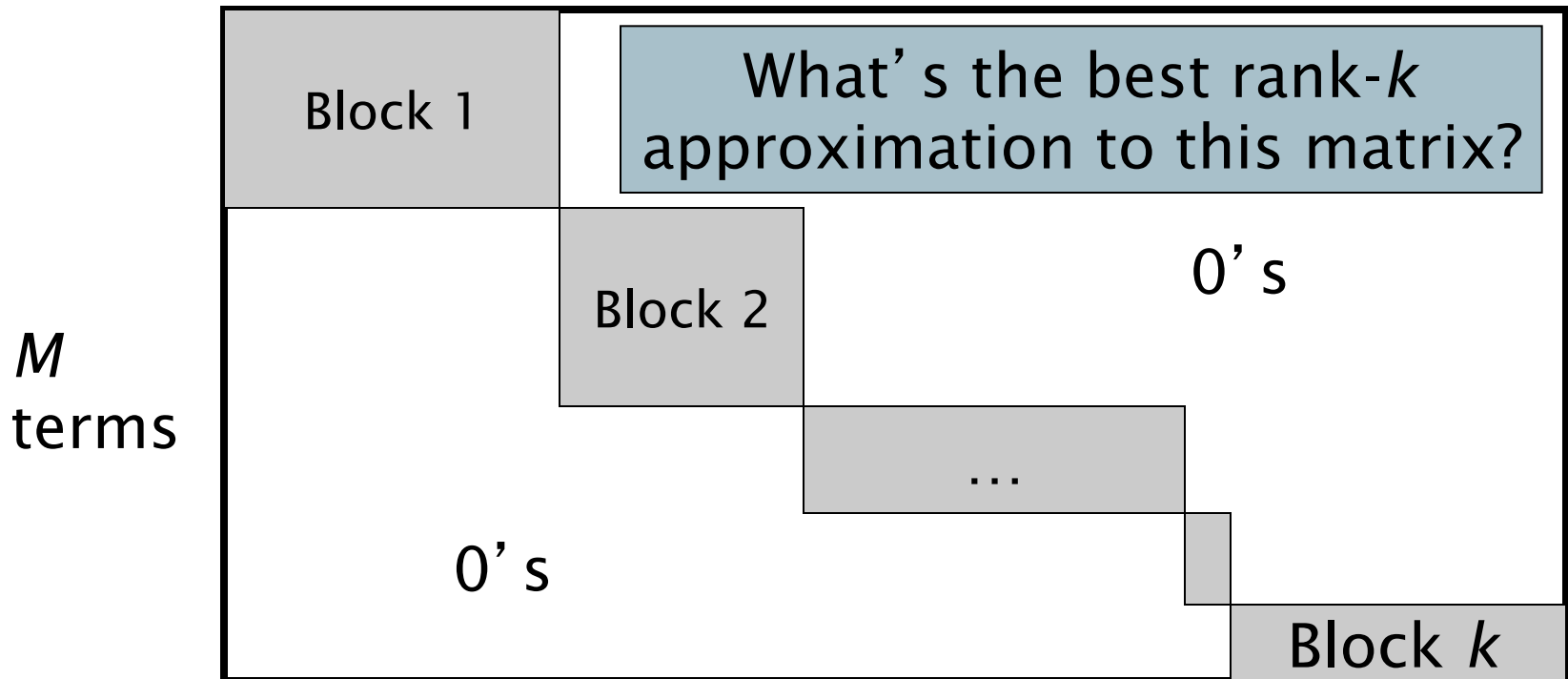
Intuition from block matrices



Vocabulary partitioned into k topics (clusters);
each doc discusses only one topic.

Intuition from block matrices

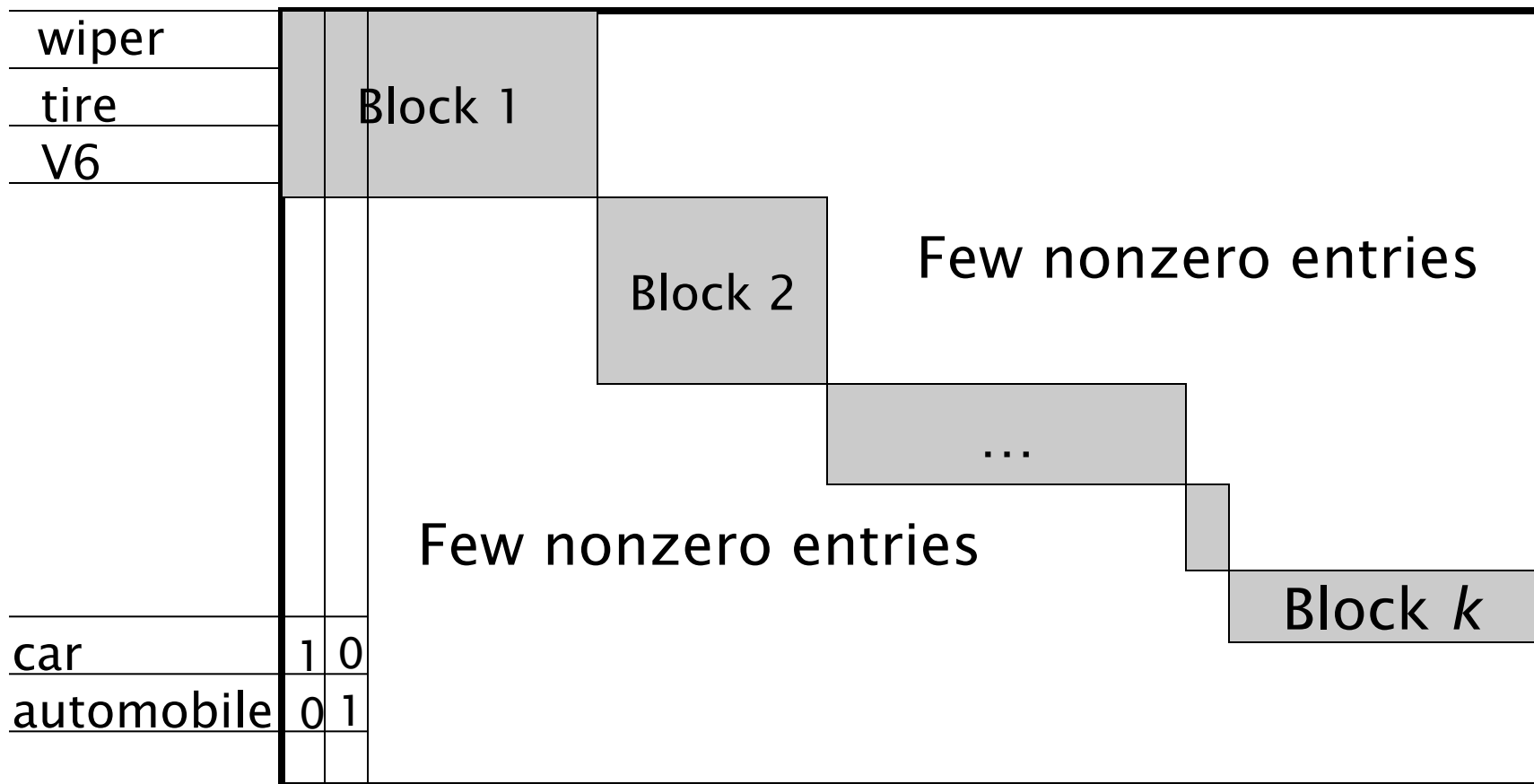
N documents



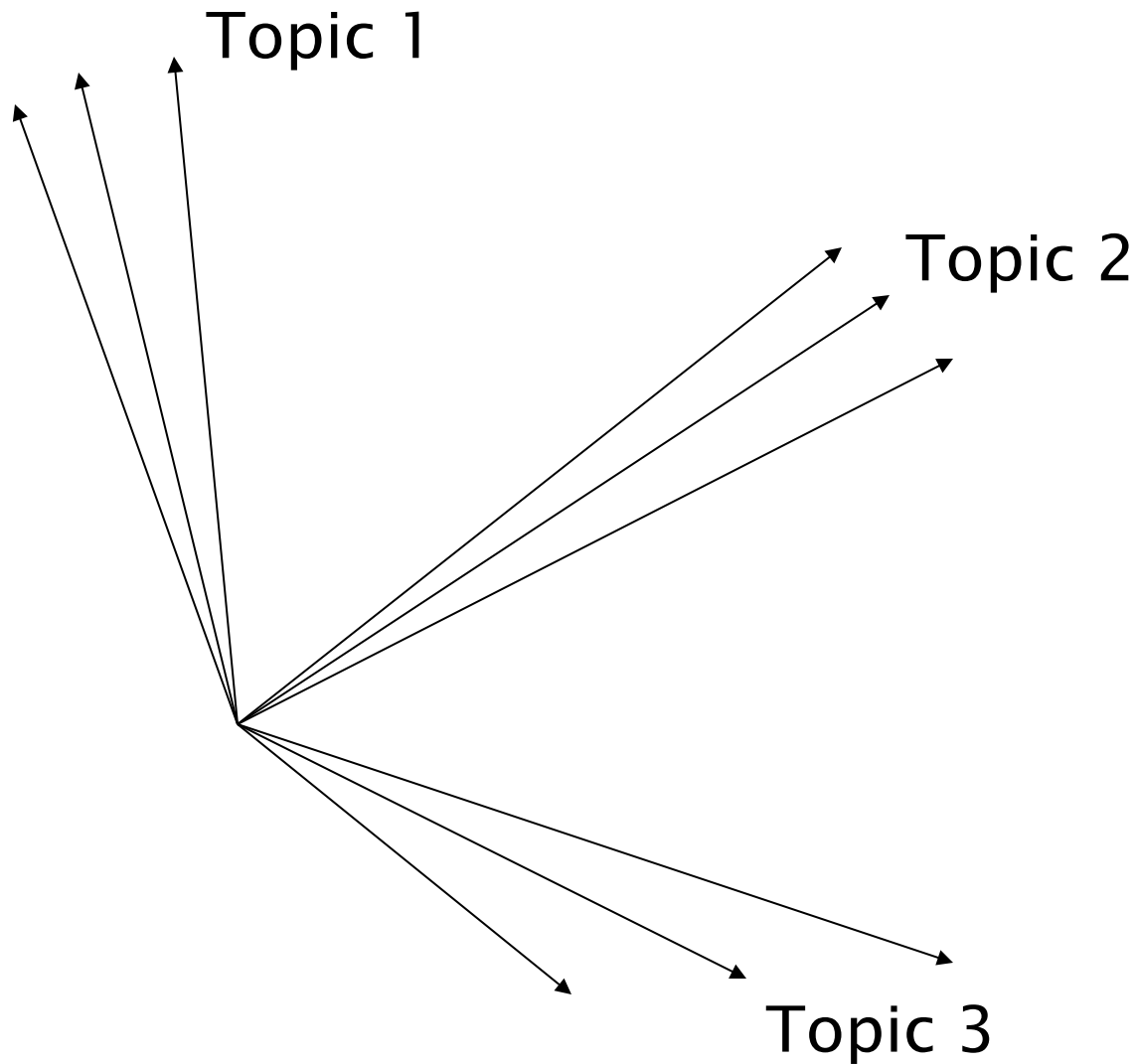
 = non-zero entries.

Intuition from block matrices

Likely there's a good rank- k approximation to this matrix.



Simplistic picture



Some wild extrapolation

- The “dimensionality” of a corpus is the number of distinct topics represented in it.
- More mathematical wild extrapolation:
 - if A has a rank k approximation of low Frobenius error, then there are no more than k distinct topics in the corpus.

LSI has many other applications

- In many settings in pattern recognition and retrieval, we have a feature-object matrix.
 - For text, the terms are features and the docs are objects.
 - Could be opinions and users ...
 - This matrix may be redundant in dimensionality.
 - Can work with low-rank approximation.
 - If entries are missing (e.g., users' opinions), can recover if dimensionality is low.
- Powerful general analytical technique
 - Close, principled analog to clustering methods.

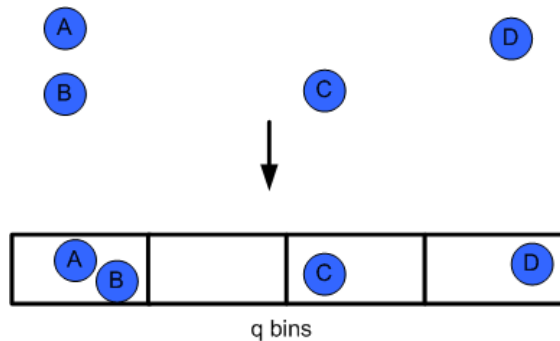
EFFICIENT SEARCH FOR CONTINUOUS REPRESENTATIONS

Document Similarity

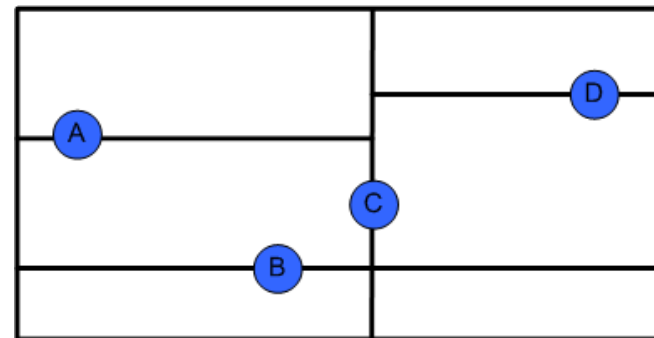
- ▶ Metric Space
 - ▶ Distance metrics:
 - ▶ Jaccard, Euclidean, Cosine, Dice, Product, etc.
- ▶ Probability Simplex
 - ▶ Information-theoretic measurements:
 - ▶ Kullback-Liebler(KL) & Jensen-Shannon(JS) divergence
 - ▶ Hellinger (He) distance
- ▶ Large Collections?
 - ▶ All-pairs comparison is practically infeasible
 - ▶ For N documents - $O(N^2)$
 - ▶ In case of a query set of k documents - $O(kN)$

Document Similarity in Large Collections

- ▶ Metric Space
 - ▶ Frame document similarity as an nearest-neighbor (NN) search problem
 - ▶ Commonly used approximate NN search approaches:
 - ▶ Locality Sensitive Hashing (LSH)
 - ▶ kd-trees



Locality Sensitive Hashing



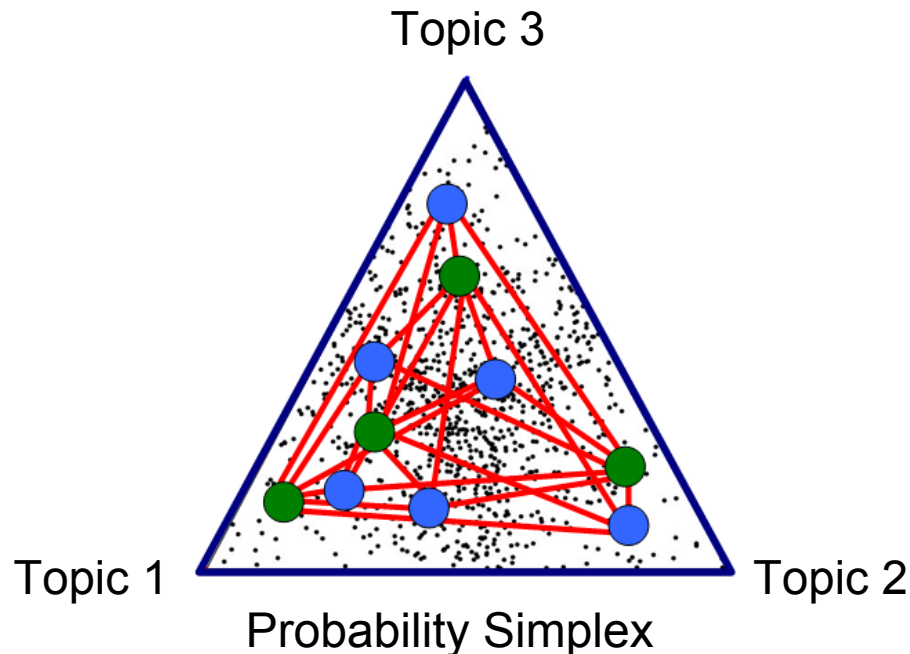
kd-trees

Document Similarity in Large Collections

- ▶ Metric Space
 - ▶ Approximate NN search approaches are designed to handle distance metrics
 - ▶ Could be directly applied to measure Euclidean, Cosine, etc.
 - ▶ But what about probability distributions?

Document Similarity in Large Collections

- ▶ Probability Simplex
 - ▶ Information-theoretic measurements
 - ▶ NN search techniques cannot be applied directly



Transforming Divergences

- ▶ Common similarity metric in the vector space model is Euclidean distance:

$$Eu(p, q) = \sqrt{\sum_{i=1}^n (p \downarrow i - q \downarrow i)^2}$$

- ▶ Compute Hellinger using Euclidean by first computing $\sqrt{p(x \downarrow i)}$ and $\sqrt{q(x \downarrow i)}$

$$He(p, q) = \sqrt{\sum_{i=1}^n (\sqrt{p(x \downarrow i)} - \sqrt{q(x \downarrow i)})^2}$$

- ▶ Discard computing the square root of the Euclidean distance
- ▶ How about Jensen-Shannon?

Transforming Divergences

- ▶ Jensen-Shannon

$$JS(p, q) = 1/2 KL(p, p + q/2) + 1/2 KL(q, p + q/2)$$

- ▶ Approximate with Hellinger?

$$He(p, q) =$$

$$He(p, q) = \sum_{i=1}^n (\sqrt{p(x_i)} - \sqrt{q(x_i)})^2$$

- ▶ Theoretical bounds do exist [Guha et. al '06]:

$$1/2 He(p, q) \leq JS(p, q) \leq 2 \ln 2 He(p, q)$$

- ▶ How tight are these bounds in practice?

Transforming Divergences

- ▶ How tight are these bounds in practice?
- ▶ Experimental Setup
 - ▶ Synthetic dataset of 100 points drawn from a Dirichlet distribution
 - ▶ Allows us to simulate collection of documents represented as multinomial distributions
 - ▶ Vary the dimensionality (D) of the multinomial
 - ▶ Different hyperparameter (α) values to vary the distribution sparsity
 - ▶ Compute Hellinger distance and Jensen-Shannon divergence across all pairs of points

Retrieving Related Patents

- ▶ Rank Aggregation
 - ▶ Xue and Croft
 - ▶ Standalone LDA representation (kd-trees)
 - ▶ Using CombSUM/CombMNZ approach

Method type	MAP	P@10	R@10
Xue and Croft	0.204	0.416	0.138
Jensen-Shannon (all pairs)	0.172	0.343	0.111
Hellinger (all-pairs)	0.178	0.345	0.112
Hellinger LSH R=0.4	0.056	0.161	0.051
Hellinger LSH R=0.6	0.091	0.248	0.078
Hellinger LSH R=0.8	0.161	0.344	0.111
Hellinger kd-trees	0.159	0.345	0.112
CombMNZ	0.232	0.442	0.145

Resources

- Manning et al., *IIR*, ch. 18
- Scott Deerwester, Susan Dumais, George Furnas, Thomas Landauer, Richard Harshman. 1990. Indexing by latent semantic analysis. *JASIS* 41(6):391—407.