variable latent semantic indexing

ravi kumar

yahoo! research

ravikumar@yahoo-inc.com

joint work with

- anirban dasgupta, cornell university
- prabhakar raghavan, yahoo! research
- andrew tomkins, yahoo! research

kdd 2005, to appear

outline

1. vector-space model
2. latent semantic indexing
3. variable latent semantic indexing

sources: berry, hofmann, langville, raghavan

thesaurus approach

luhn (1957, 1961)

- **similar/related words grouped into notional families**
- **represent document by notional elements (vocabulary based)**
- **matching by measuring degree of notional similarity**
- **key word in context indexing (kwic); technical manuals**

quick **brown** fox	line 1
lazy **dog**	…
brown **fox** jumps	…
fox **jumps** over	…
the **lazy** dog	…

lattices of search terms

joyce, needham (1958)

- lattices and hierarchies of search terms
  - thesaurus/dictionary
- if a term occurs in a document, then all all terms present at higher levels of the lattices are also considered occurring
  - punch cards!

term associations

doyle (1962)

- identify unusual co-occurrences of pairs of words
  - term associations
- statistical methods used to identify correlations
  - chi-square tests, pearson correlation coefficient
- term association maps for interactive retrieval

- semantic maps
  - natural language understanding

vector-space model

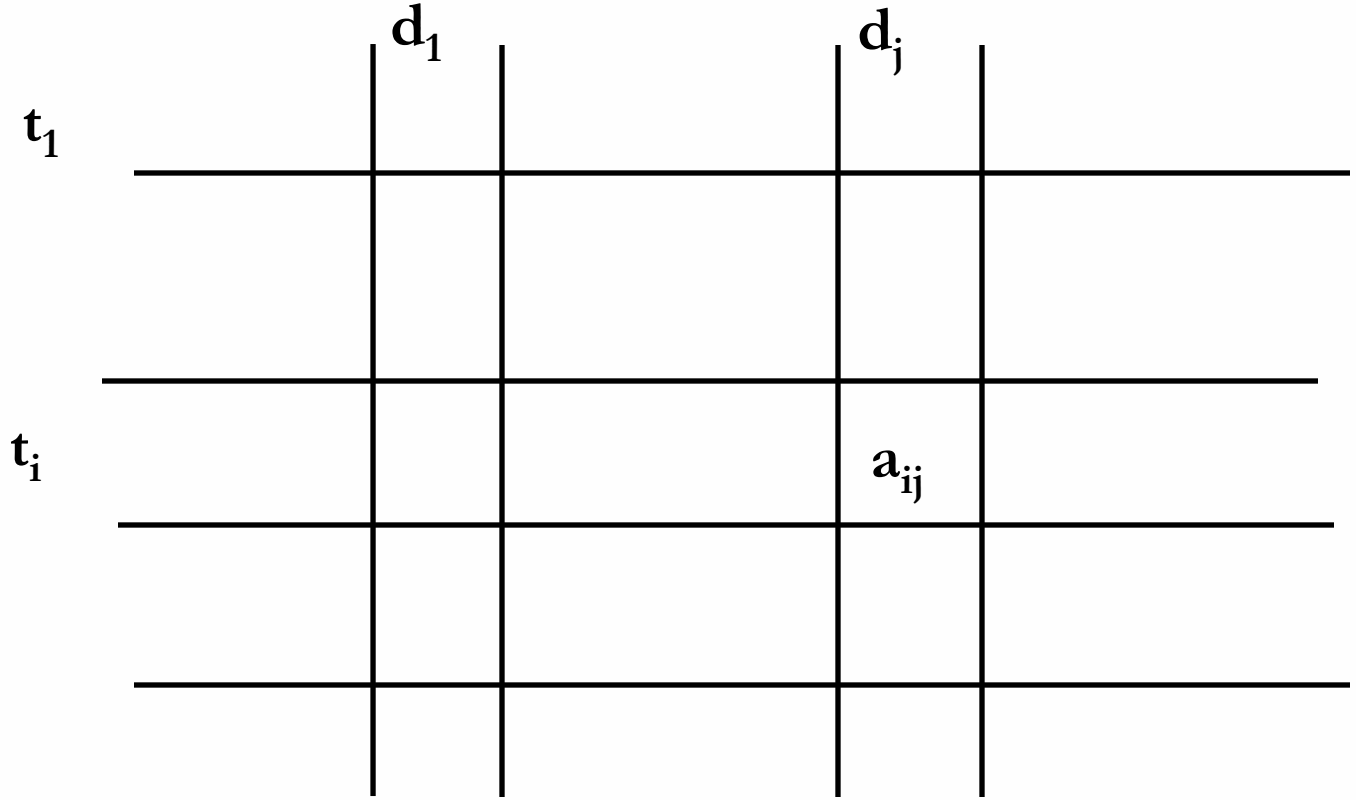
salton (1960s)

- each dictionary term is a dimension
- each document is a high-dimensional vector
- entries can be boolean or weighted
- geometric interpretation for similarity, etc

- postulate: documents close in vector space talk about same things
- can implement "query-by-example"

term-document matrix

- D = { d_1, … } = set of documents in the collection
- T = { t_1, … } = set of terms



selecting terms

pseudo-linguistic heuristics

- drop stop-words

  eg, articles, “html”, …
- stemming/lemmatization

  eg, speaks, speaking α speak
- use only nouns/noun phrases

  people, places, entity extraction
- phrase extraction

  eg, “university of california”
- domain-specific knowledge

similarity measure

similarity between document and query

euclidean distance: sensitive to vector length

angular cosine measure

$sim(d, q) = \cos \text{Å}(d, q) = \text{h}d, q\text{i}/(\text{k}d\text{k} \text{¢} \text{k}q\text{k})$

given a query

compute $\delta_i = sim(d_i, q)$

rank documents according to δ_i

retrieve top documents

example from berry's book

terms

t_1: bab(y,ies,y's)

t_2: child(ren's)

t_3: guide

t_4: health

t_5: home

t_6: infant

t_7: proofing

t_8: safety

t_9: toddler

documents

d_1: infant & toddler first aid

d_2: babies and children's room (for your home)

d_3: child safety at home

d_4: your baby's health & safety: from infant to toddler

d_5: baby proofing basics

d_6: your guide to easy rust proofing

d_7: beanie babies collector guide

eg, query = baby health

$$\mathbf{A} = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \end{array} \begin{array}{c} \begin{array}{ccccccc} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \\ \delta_7 \end{bmatrix} = \begin{bmatrix} 0 \\ .5774 \\ 0 \\ .8944 \\ .7071 \\ 0 \\ .7071 \end{bmatrix}$$

answer = d4: your baby's health & safety: from infant to toddler

smart system

system for mechanical analysis and retrieval of text

aka

salton's magical automatic retriever of text

term-weighting

- document length normalization
- global term weights
  - boolean
  - tf-idf
  - okapi

advantages of vector-space model

- no subjective selection of index terms
- partial match
  - no document contains all search terms
- highly automatable
  - similarity scores
- flexible term-weighting schemes
- extensions
  - document clustering (doc-doc similarity)
  - term clustering
  - relevance feedback
- geometric foundation

synonymy

synonymy: different terms with similar meaning

- car vs automobile
- words indicating the same topic
- vector space model does not identify them together
- context can be used to bring them together

… car …
… road …
… accident …
… hurt …
… gas …

… road …
… automobile …
… gas …
… injury …
… accident …

polysemy

polysemy: same word with different meanings depending on context

- jaguar, windows, apple
- esp severe in heterogeneous collection of documents
- vector space model does not discriminate them
- context can be used to differentiate them

… hunts …
… jaguar …
… wild …
… nocturnal …

… expensive …
… mph …
… jaguar …
… car …

drawbacks of vector space model

- dimensionality
  - representation of documents is in very high dimension
  - curse of dimensionality for learning/estimation
- document vectors
  - sparse
  - cosine similarity could be noisy
- syntax
  - bag-of-words
  - phrases/order-relationships missed out
- semantics
  - only exact matches identified
  - semantic relations between words not exploited

latent semantic indexing

latent semantic indexing

- 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
  - similar terms map to similar location in low dimensional space
  - noise reduction by dimension reduction

dim. reduction vs. clustering

- common use of dimension reduction
  - find "better" representation of data
    - supporting more accurate retrieval
    - supporting more efficient retrieval
  - still using all points, but in a new representational space
- common use of clustering
  - summarize data or reduce data to fewer objects
  - clusters are often first-class citizens, directly used in the user interface or as part of retrieval algorithm

random projections

take vectors and pack them into small number of dimensions

- choose a random direction x_1 in the vector space
- for i = 2 to k

  choose random direction x_i ⊥ $x_1, x_2, \ldots x_{i-1}$
- project each document vector into the subspace spanned by $\{x_1, x_2, \ldots, x_k\}$

johnson-lindenstrauss theorem: with high probability, relative distances are (approximately) preserved by projection

linear algebra recap

eigenvalues and eigenvectors

eigenvectors, for a square mxm matrix S

$$S\,v = \lambda\,v$$

(right) eigenvector $v \in R^m \neq 0$

eigenvalue $\lambda \in R$

example

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

$$Sv = \lambda v \text{ , } (S - \lambda I)v = 0$$

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

this is an m-th order equation in λ that can have at most m distinct solutions (roots of the characteristic polynomial) – can be complex even though S is real

matrix-vector multiplication

$$S = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

has eigenvalues 3, 2, 0 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 can be viewed as a linear combination of the eigenvectors eg, $(2\ 4\ 6)^t = 2v_1 + 4v_2 + 6v_3$

matrix-vector multiplication

- thus a matrix-vector multiplication such as Sx can be rewritten in terms of the eigenvalues/vectors

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

$$Sx = 2Sv_1 + 4Sv_2 + 6Sv_3 = 2\lambda_1 v_1 + 4\lambda_2 v_2 + 6\lambda_3 v_3$$

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

- suggestion: the effect of "small" eigenvalues is small

facts: eigenvalues, eigenvectors

for symmetric matrices, eigenvectors for distinct eigenvalues are orthogonal

$$Sv_1 = \lambda_1 v_1, Sv_2 = \lambda_2 v_2, \lambda_1 \neq \lambda_2 \Rightarrow \langle v_1, v_2 \rangle = 0$$

all eigenvalues of a real symmetric matrix are real

all eigenvalues of a positive semi-definite matrix are non-negative

$$\forall w \in \mathbb{R}^n, w^t S w \geq 0, \text{ then if } Sv = \lambda v \Rightarrow \lambda \geq 0$$

example

- let $S = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ ← real, symmetric

- the eigenvalues are 1 and 3 (nonnegative, real)
- the eigenvectors are orthogonal (and real)

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

eigen/diagonal decomposition

let $S \in R^{m \times m}$ be a square matrix with m linearly independent eigenvectors (a "non-defective" matrix)

- matrix diagonalization theorem: exists an eigen decomposition

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

- columns of U are eigenvectors of S
- diagonal elements of Λ are eigenvalues of S

$$\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n), \lambda_i \geq \lambda_{i+1}$$

proof of theorem

let U have the eigenvectors as columns $U = [v_1, \ldots, v_n]$

then, SU can be written

$$S\,U = S\,[v_1, \ldots, v_n] = [\lambda_1 v_1, \ldots, \lambda_n v_n]$$
$$= [v_1, \ldots, v_n] \cdot \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$$

thus

$$SU = U\Lambda$$

or

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

example

recall $S = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}; \lambda_1 = 1, \lambda_2 = 3. \quad U = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$

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

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

eg, contd

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

then, S =

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

Q Λ ($Q^{-1} = Q^t$)

symmetric eigen decomposition

if $S \in R^{m \times m}$ is a symmetric matrix

theorem: exists a (unique) eigen decomposition

$$S = Q \Lambda Q^t$$

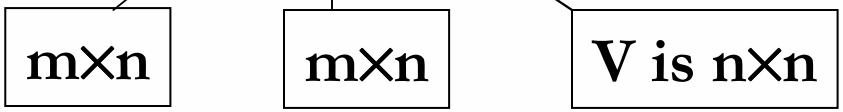
where Q is orthogonal

- $Q^{-1} = Q^t$
- columns of Q are normalized eigenvectors
- columns are orthogonal
- (everything is real)

singular value decomposition

for an m× n matrix **A** of rank r there exists a unique factorization (singular value decomposition = svd)

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



columns of U are orthogonal eigenvectors of AA^t

columns of V are orthogonal eigenvectors of A^tA

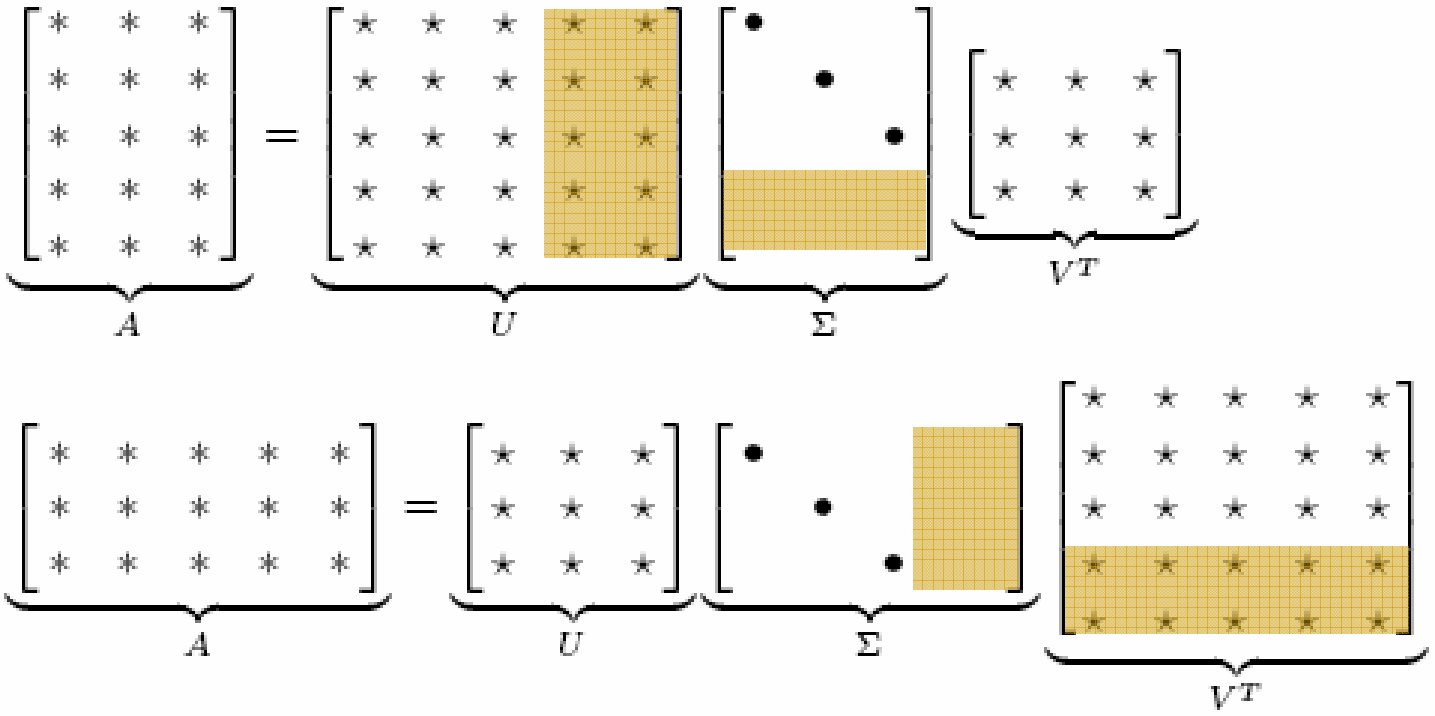
eigenvalues $\lambda_1 \ldots \lambda_r$ of AA^T are the eigenvalues of A^TA

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

$$\Sigma = \text{diag}(\sigma_1, \ldots \sigma_r, 0, \ldots, 0) \text{ (singular values}$$

singular value decomposition

illustration of svd dimensions and sparseness



svd example

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

thus m=3, n=2 and 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 A_k of rank k such that

$$A_k = \min_{X:\ \mathrm{rank}(X) = k} \|A - X\|_f$$

frobenius norm: $\|A\|_f^2 = \sum_{i,j} |a_{ij}^2|$

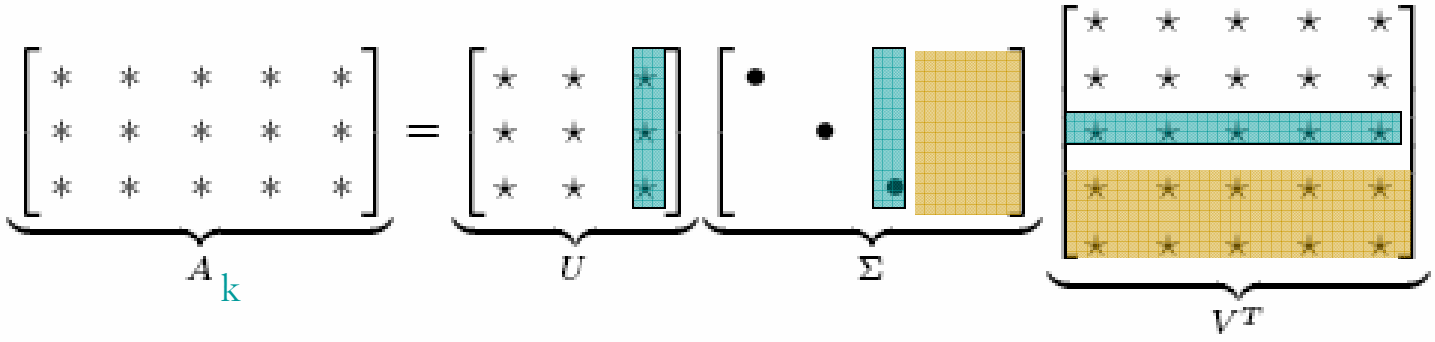
A_k and X are both m×n matrices

typically, want k ¿ r

low-rank approximation

solution via svd

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



$$A_k = U_k\ \Sigma_k\ V_\kappa^t = \sum_{i=1,k}\ \sigma_i\ u_i\ v_i^t = \text{sum of rank 1 matrices}$$

approximation error

how good (bad) is this approximation?

it is the best possible, measured by the frobenius norm of the error

$$\min_{X:\ \mathrm{rank}(X) = k} \|A - X\|_f = \|A - A_k\|_f = \sqrt{\sum_{j > k} \sigma_j^2}$$

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

suggests why frobenius error drops as k increases

eckart-young theorem

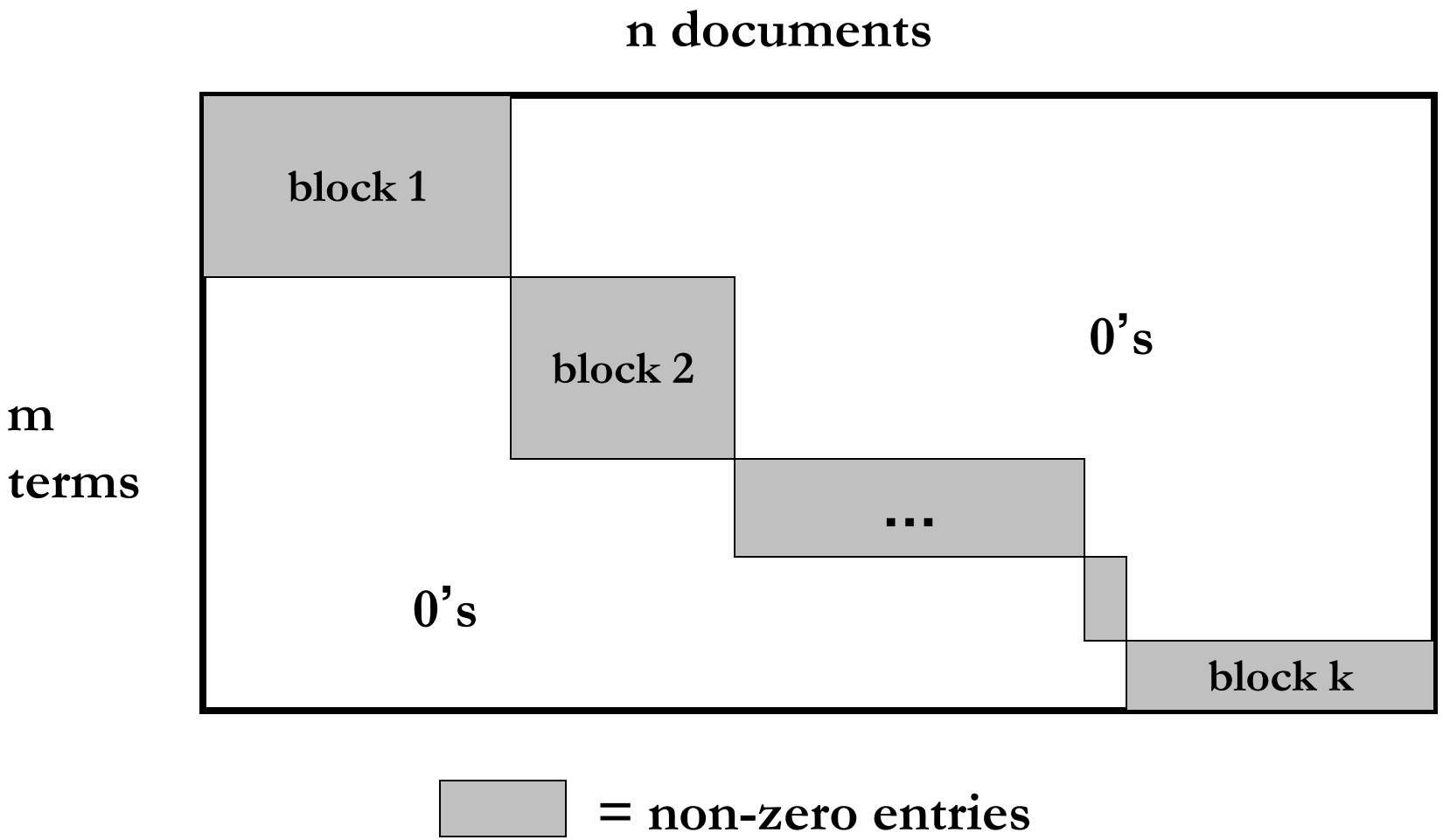
svd vs random projections

- completely different method for low-rank approximation
- random projection is data-oblivious
  - svd-based approximation is data-dependent
- error for random projection depended only on start/finish dimensionality
  - for every distance
- error for svd-based approximation is for the frobenius norm, not for individual distances
- eliminate redundant axes, brings relevant dimensions together

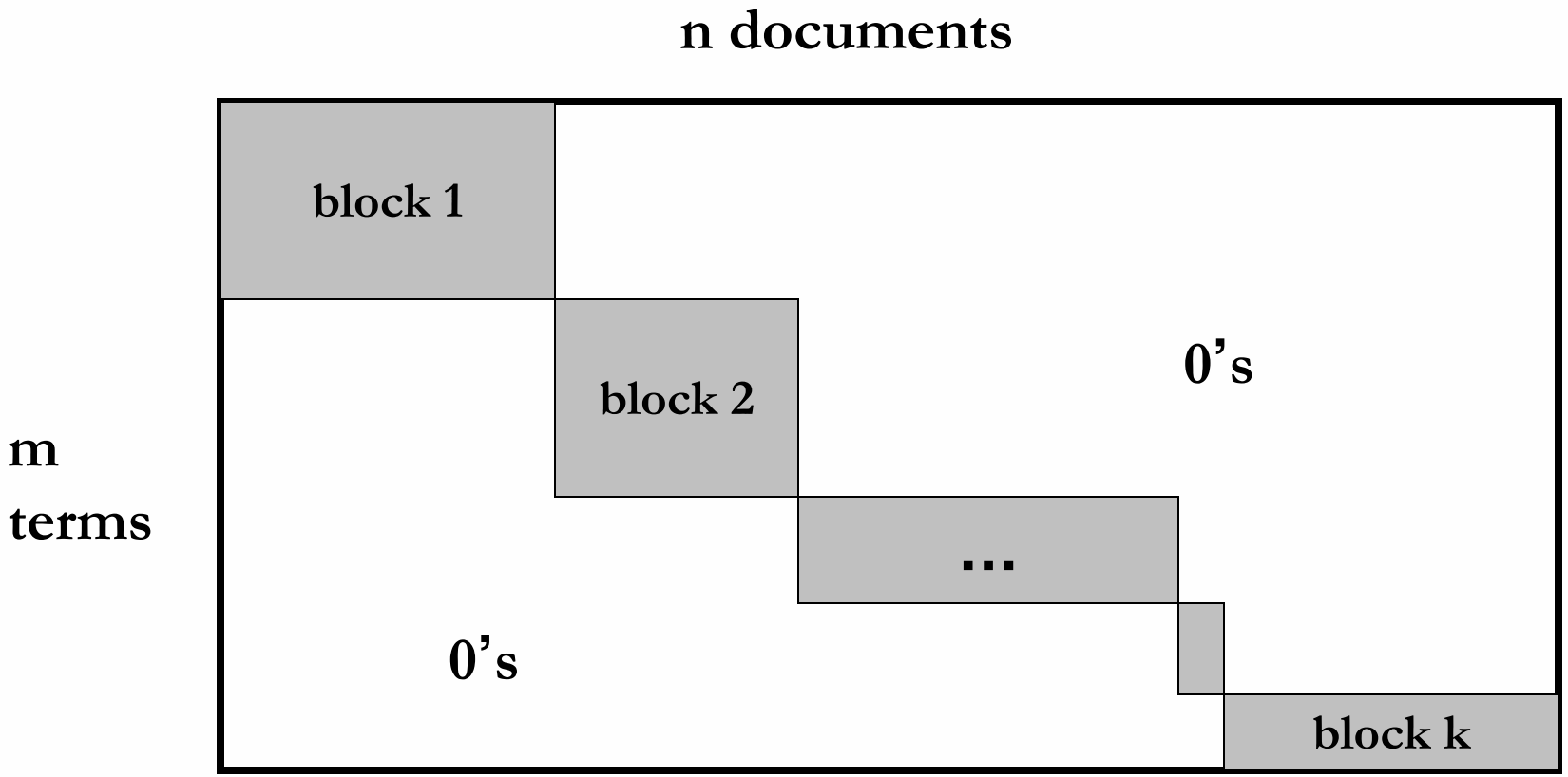
latent semantic indexing (lsi)

- from term-document matrix A, compute the approximation A_k
- there is a row for each term and a column for each document in A_k
- thus documents live in a space of k ¿ r dimensions
  - these dimensions are not the original axes
- query mapped to this space
  - not a sparse vector
- compute document similarity based on the inner product in this latent semantic space

intuition from block matrices

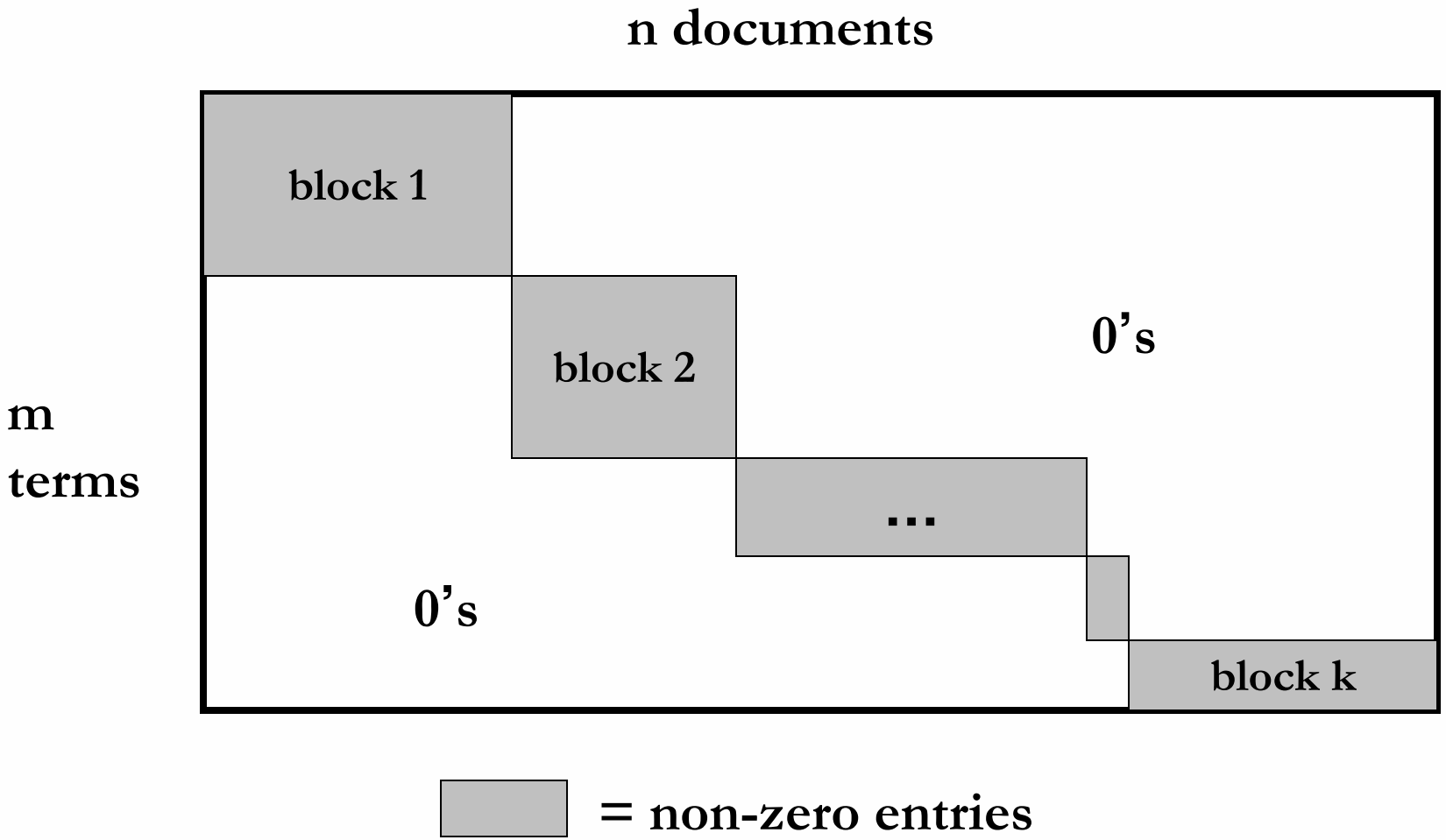


intuition, contd

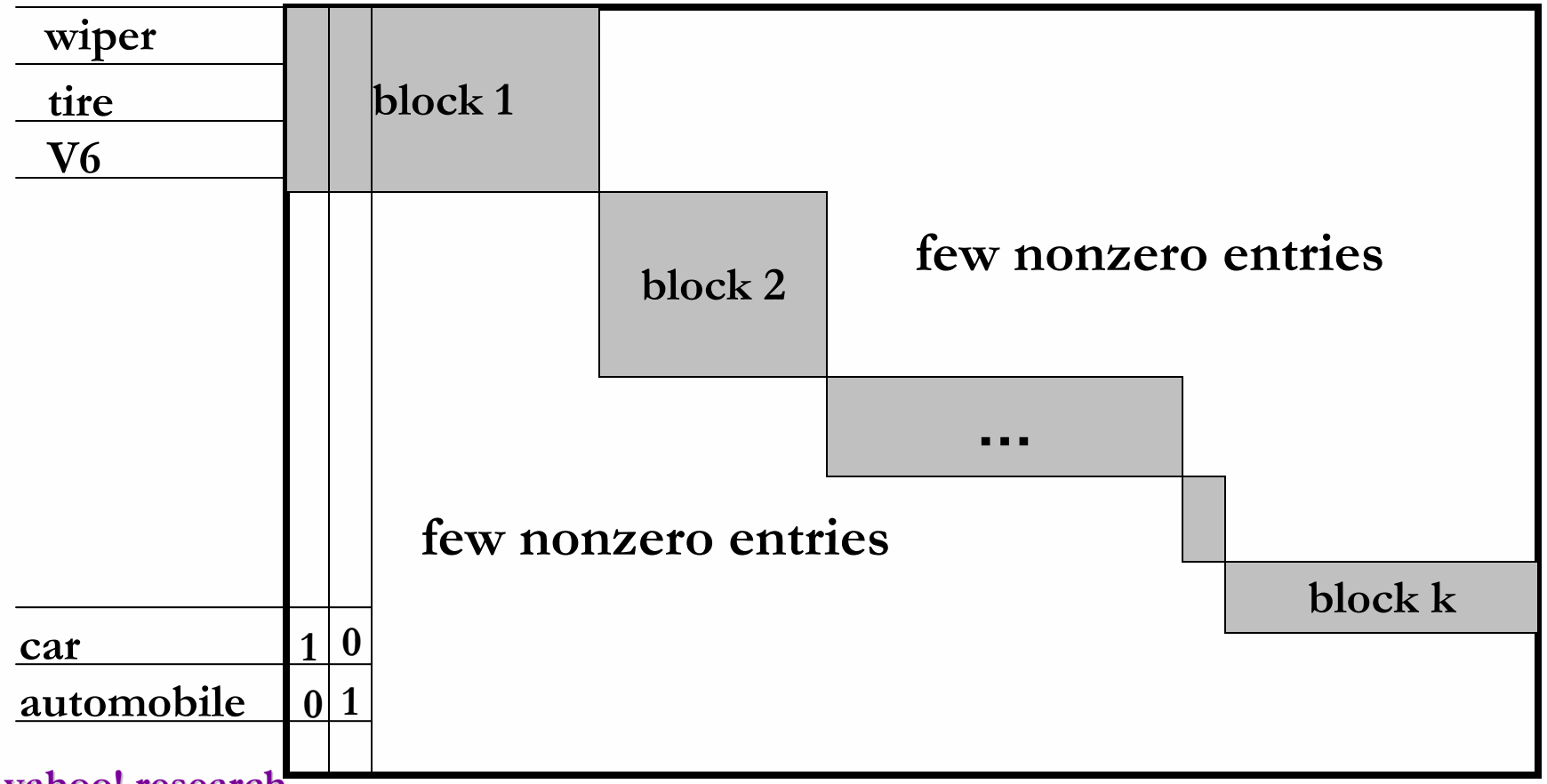


vocabulary partitioned into k topics (clusters); each document discusses only one topic

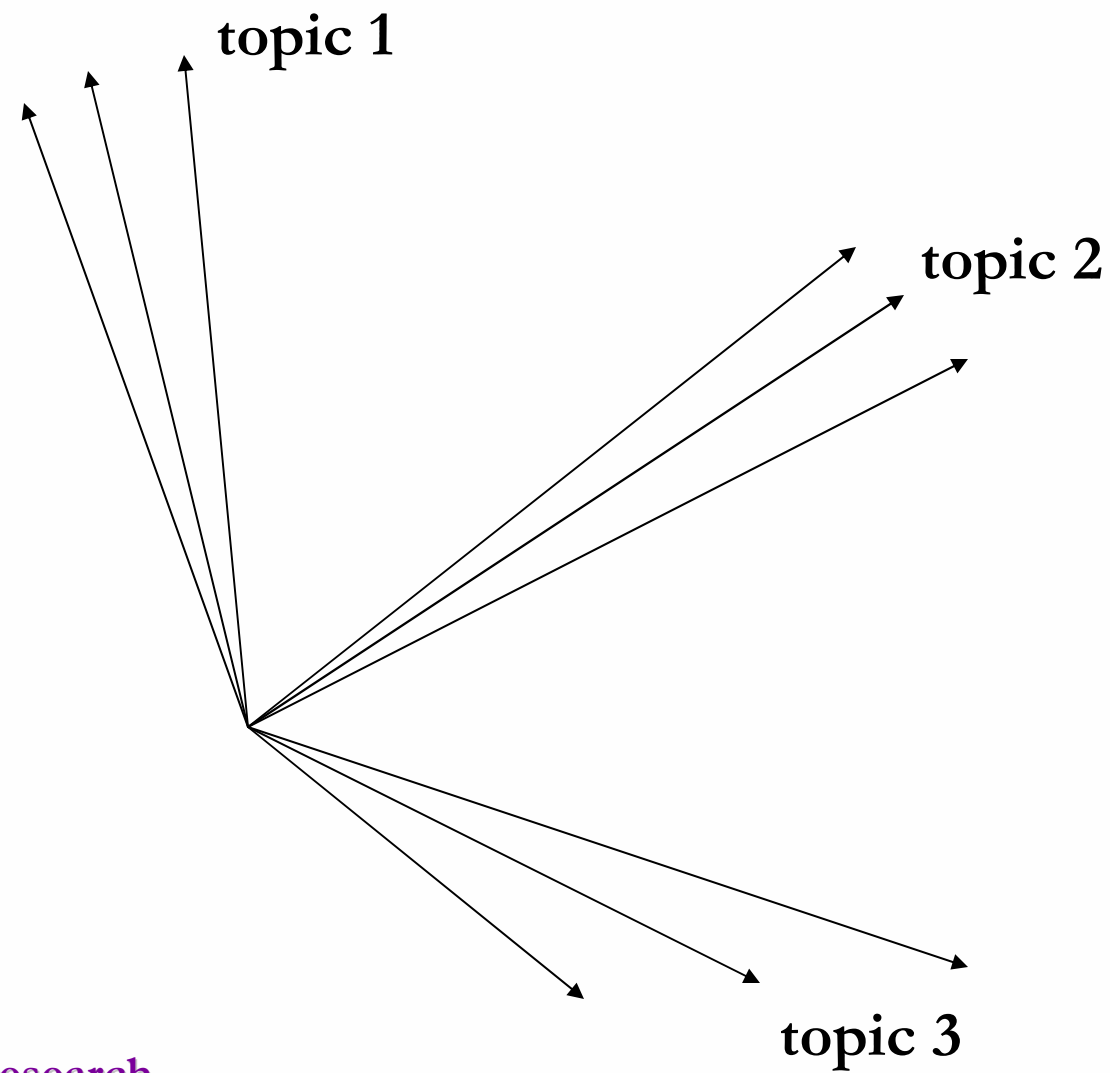
intuition, contd



intuition, contd



simplistic picture



some wild extrapolation

"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

empirical evidence

- experiments on trec 1/2/3 – dumais
- lanczos svd code (available on netlib) due to berry used in these expts
  - running times of ~ one day on tens of thousands of docs
- dimensions – various values 250-350 reported
  - (under 200 reported unsatisfactory)
- generally expect recall to improve – what about precision?

empirical evidence, contd

- 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

variable latent semantic indexing

low dimensional approx, recap

- extracts salient features from corpus
- low-dimensional representation
  - storage efficiency
  - computational efficiency
- disambiguation based on context
- "clustering" effect
  - brings relevant documents together
- noise reduction

effect of queries

standard lsi is oblivious to queries

200-300 dimensions needed to get reasonable errors

suppose most of the queries come from finance, sports

can we tune lsi against this query distribution?

adhoc: throw out non-finance, non-sport terms from dictionary and compute lsi

principled approach?

linear algebra, recap

- svd

  $A = U \Sigma V^t$

  U, V column orthogonal, Σ diagonal

- symmetric psd eigen decomposition

  $C = Y \Lambda Y^t$

  Y orthogonal, $\Lambda \succeq 0$

- frobenius norm

  $\|A\|_f^2 = \sum_{i,j} a_{ij}^2 = \mathrm{Tr}\, A^t A$

- svd low-rank approximation

  $A_k = \sum_{i=1,k} \sigma_i u_i v_i^t$

query distribution, co-occurrence

Q = distribution on terms

co-occurrence matrix

$C = C_Q = E_{q \sim Q} [q\ q^t]$

fact: C is positive semi-definite

proof: $v^t\ C\ v = E[v^t\ q\ q^t\ v] = E[(v^t\ q)^2] \geq 0$

corollary: $C = Y\ \Lambda\ Y^t$

svd and average distortion

suppose Q is such that each coordinate is iid unit normal, ie, C = I

find rank k matrix X that minimizes the average distortion imparted to random vector according to Q

$$E_{q \sim Q} [\mathbf{k}\, q^t (A - X)\, \mathbf{k}_2^2]$$

answer: $X = A_k$

proof when C = I

$$E[\|k\, q^t (A - X)\|_2^2]$$

$$= E[q^t (A - X)(A^t - X^t)\, q]$$

$$= E[Tr\,((A - X)qq^t(A^t - X^t))]$$

$$= Tr\,((A^t - X^t)\, ¢\, C\, ¢\, (A - X))$$

$$= Tr\,((A - X)^t (A - X))$$

$$= \|kA - X\|_f^2$$

rotating the space

assume C is full rank and define

$$C^{1/2} = Y \Lambda^{1/2} Y^T$$

$$C^{-1/2} = Y \Lambda^{-1/2} Y^T$$

define $w = C^{-1/2} q$

fact: $E[ww^t] = I$

proof: $E[ww^t] = E[C^{-1/2} qq^t C^{-1/2}]$

$= C^{-1/2} ¢ C ¢ C^{-1/2}$

$= I$

main idea

- rotate the space of query vectors so that the query distribution becomes isotropic
- find the svd of term-document matrix in the rotated space
- compute the low-rank approximation
- rotate the low-rank approximation back to the original space

some details

$$E[\|q^t (A - X)\|^2]$$

$$= E[\|w^t C^{1/2} (A - X)\|^2]$$

$$= E[\|w^t (C^{1/2} A - C^{1/2} X)\|^2]$$

minimized when

$$C^{1/2} X = \text{svd}(C^{1/2} A)$$

and so

$$X = C^{-1/2} \text{svd}(C^{1/2} A)$$

main result

alternate statement

if V_k = top k right singular vectors of $C^{1/2} A$ in its columns, then the minimizing solution is $A V_k V_k^t$

space requirements = same as svd

store $A V_k$ and V_k^t

experimental setup

- reuters data (1987)
  - 21k documents
  - five categories
  - 112k terms
  - 134 terms/document
- preprocessing
  - porter stemmed, case folded, stop worded
  - term-document matrix with boolean, okapi weighting
- svdpackc
  - external memory implementation

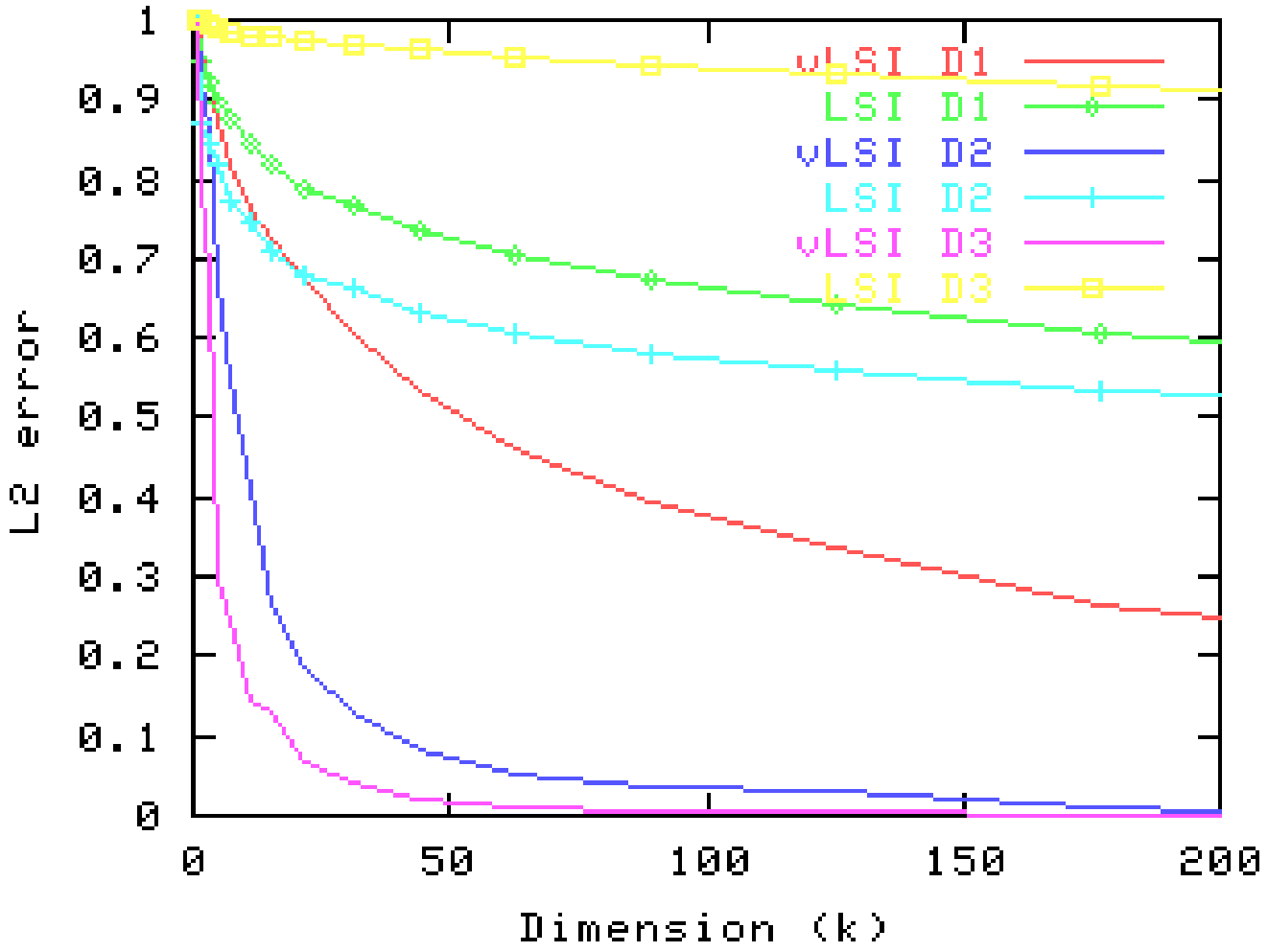
metrics

- l_2 error
  - $\|q^t (A - A_k)\|$
- competitive precision
  - S = top k documents ranked by vlsi
  - cp(d) = $|S \cap A[1\ldots d]|/d$

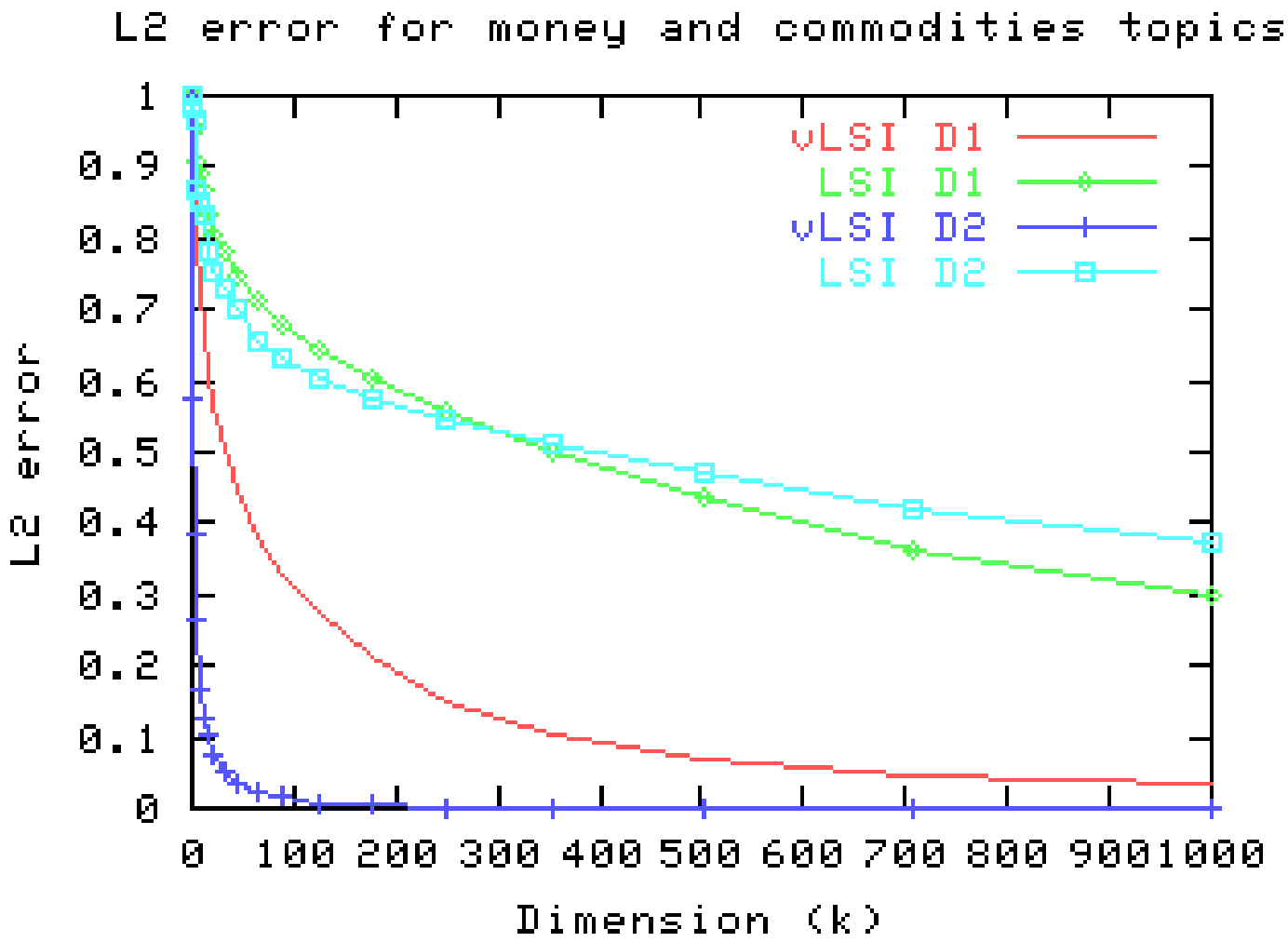
query distributions

- single word
  - terms distributed according to distribution in corpus
  - terms distributed power law, ordered by distribution in corpus
  - terms distributed by power law, ordered randomly
- two topics
  - money, commodity
- double word
  - top two bigrams

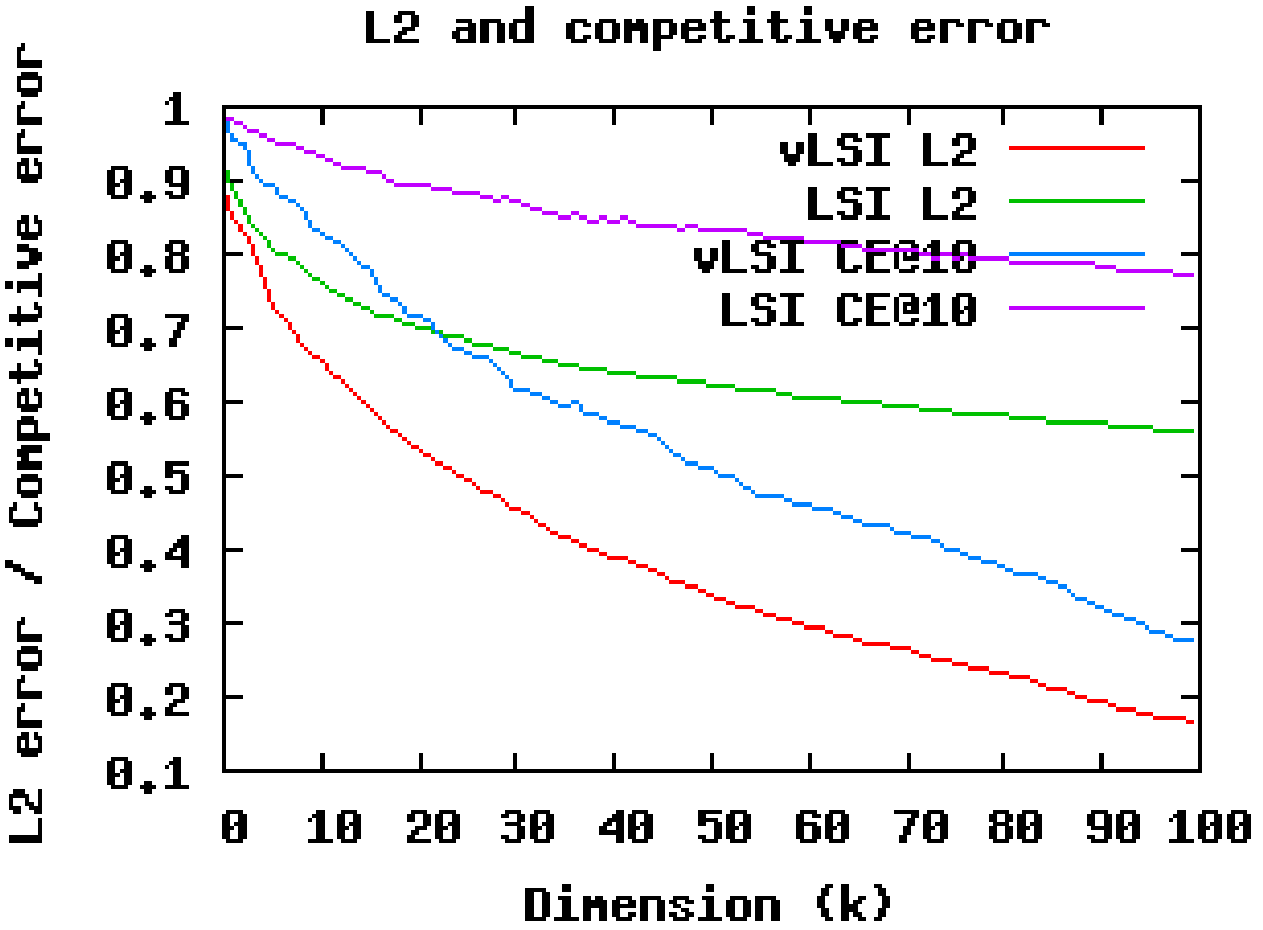
results, l2 error



results, money, l2 error



results, multiword queries



future work

- personalized version
- computational questions
  - efficiency
  - approximate
  - sampling
  - data stream
- perturbation analysis
- beyond text domains

thank you!