# Roadmap

- Motivation
- Matrix tools
- Tensor tools
- Case studies

- SVD, PCA
- HITS, PageRank
- CUR
- Co-clustering

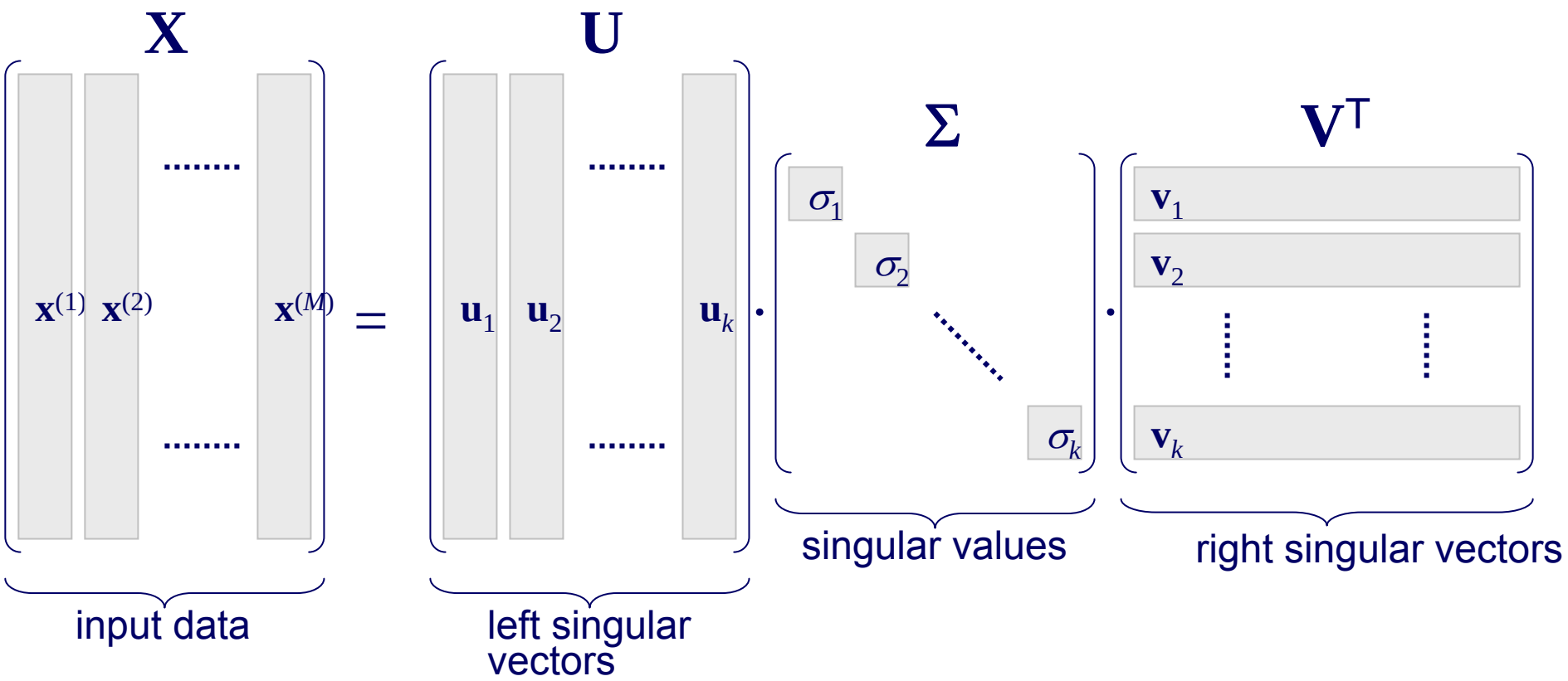
# Examples of Matrices

- Example/Intuition: Documents and terms
- Find patterns, groups, concepts

| | data | mining | classif. | tree | ... |
|---|---|---|---|---|---|
| Paper#1 | 13 | 11 | 22 | 55 | ... |
| Paper#2 | 5 | 4 | 6 | 7 | ... |
| Paper#3 | ... | ... | ... | ... | ... |
| Paper#4 | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... |

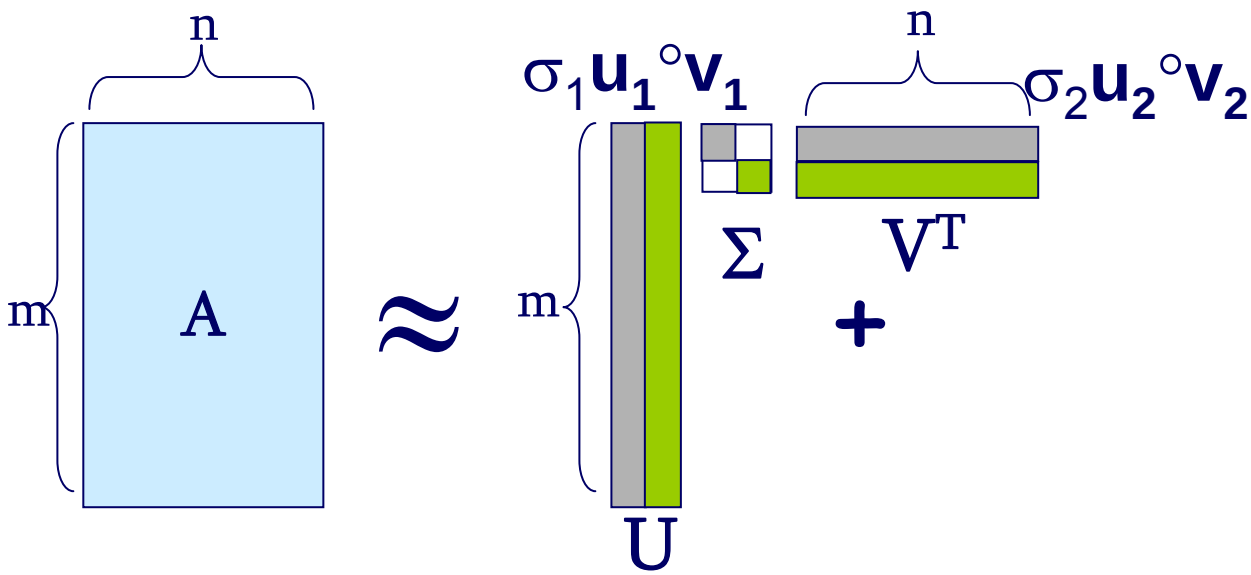
# Singular Value Decomposition (SVD)

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^{\mathsf{T}}$$



# SVD as spectral decomposition

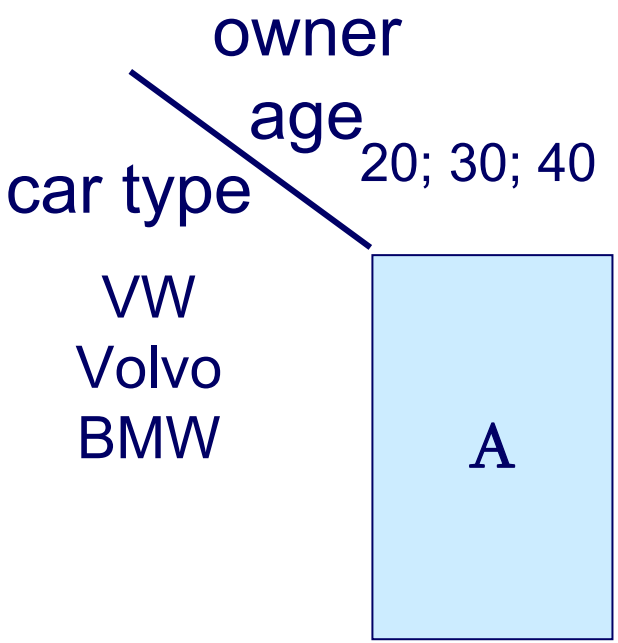
$$\mathbf{A} \approx \mathbf{U}\Sigma\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i$$

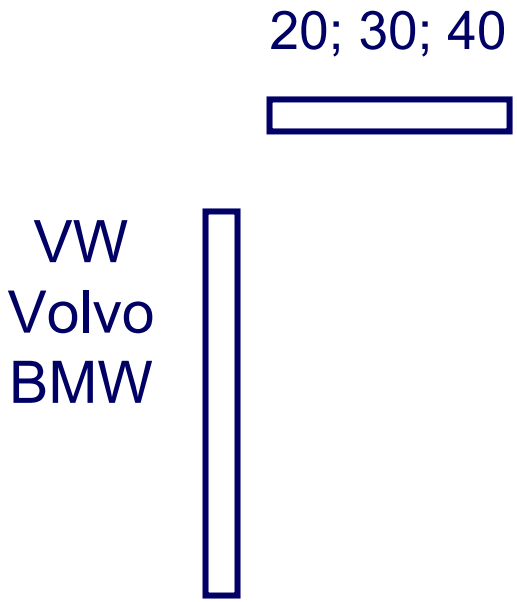



- Best rank-k approximation in L2 and Frobenius
- SVD only works for static matrices (a single 2nd order tensor)

See also PARAFAC

# Vector outer product – intuition:



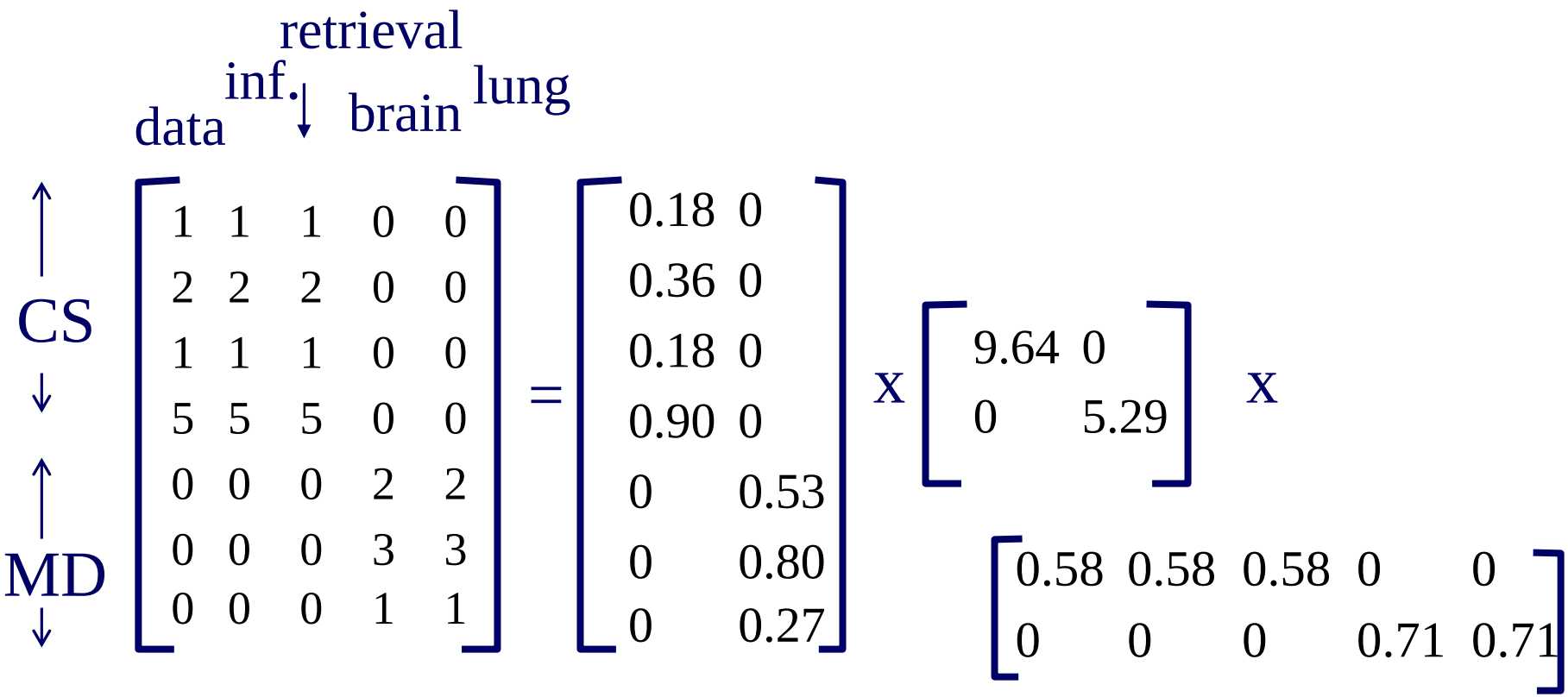




2-d histogram

1-d histograms + independence assumption

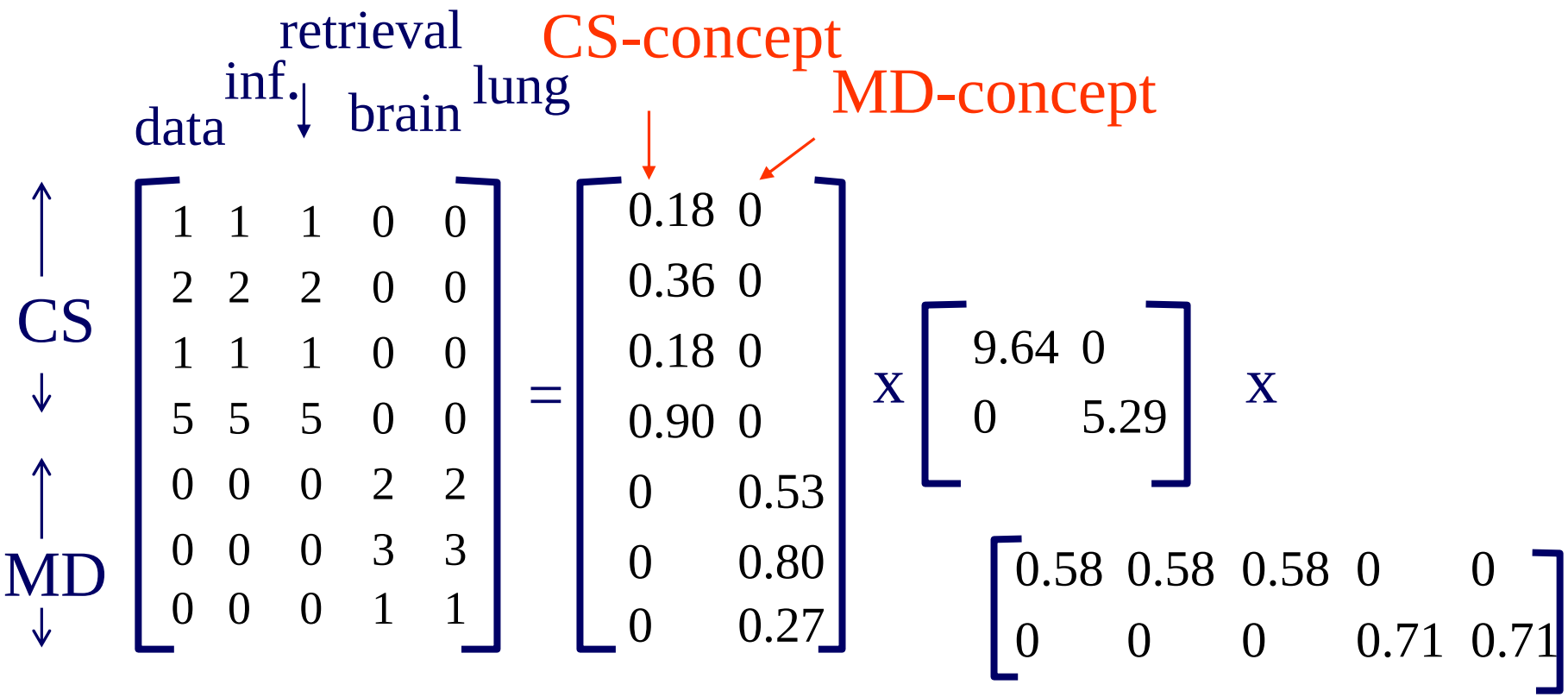
# SVD - Example

- **A = U Σ $V^T$** - example:



# SVD - Example

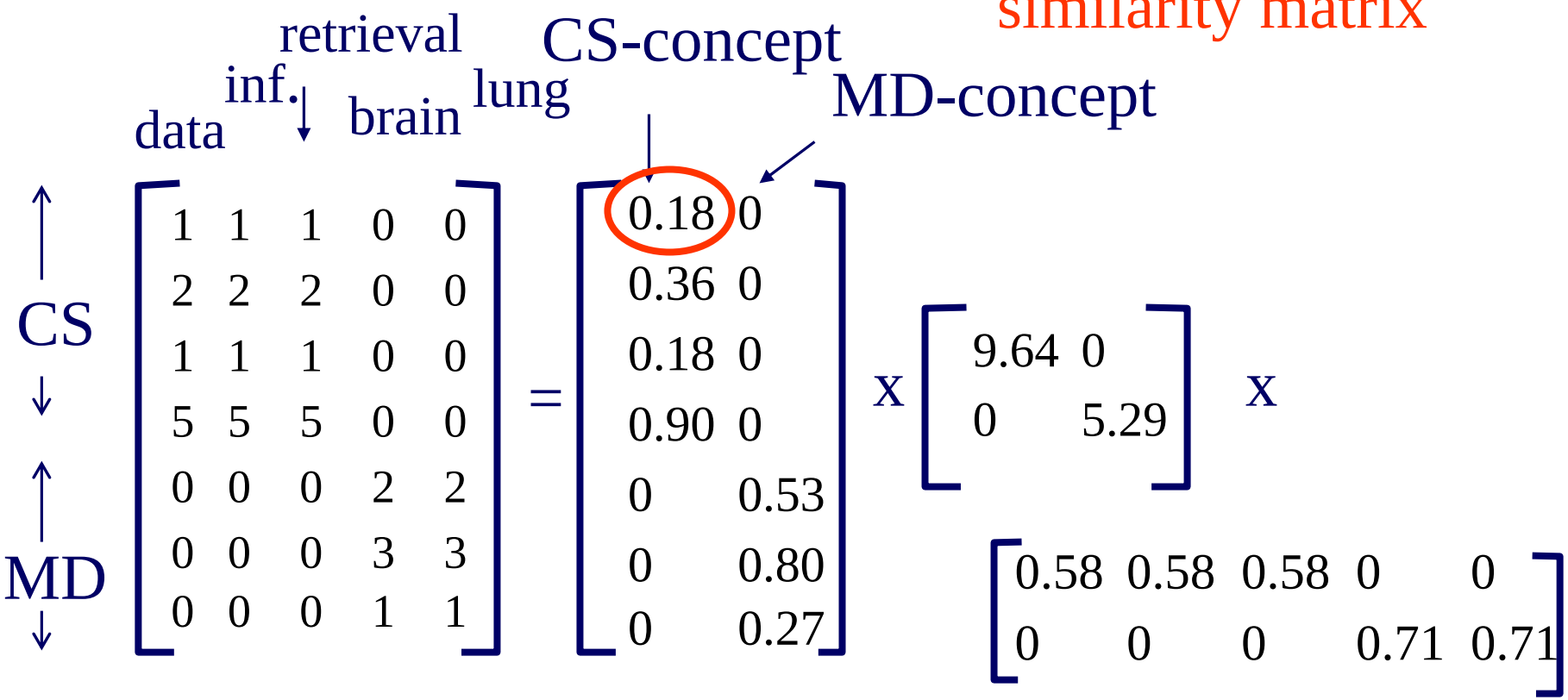
- **A = U Σ $V^T$** - example:



# SVD - Example

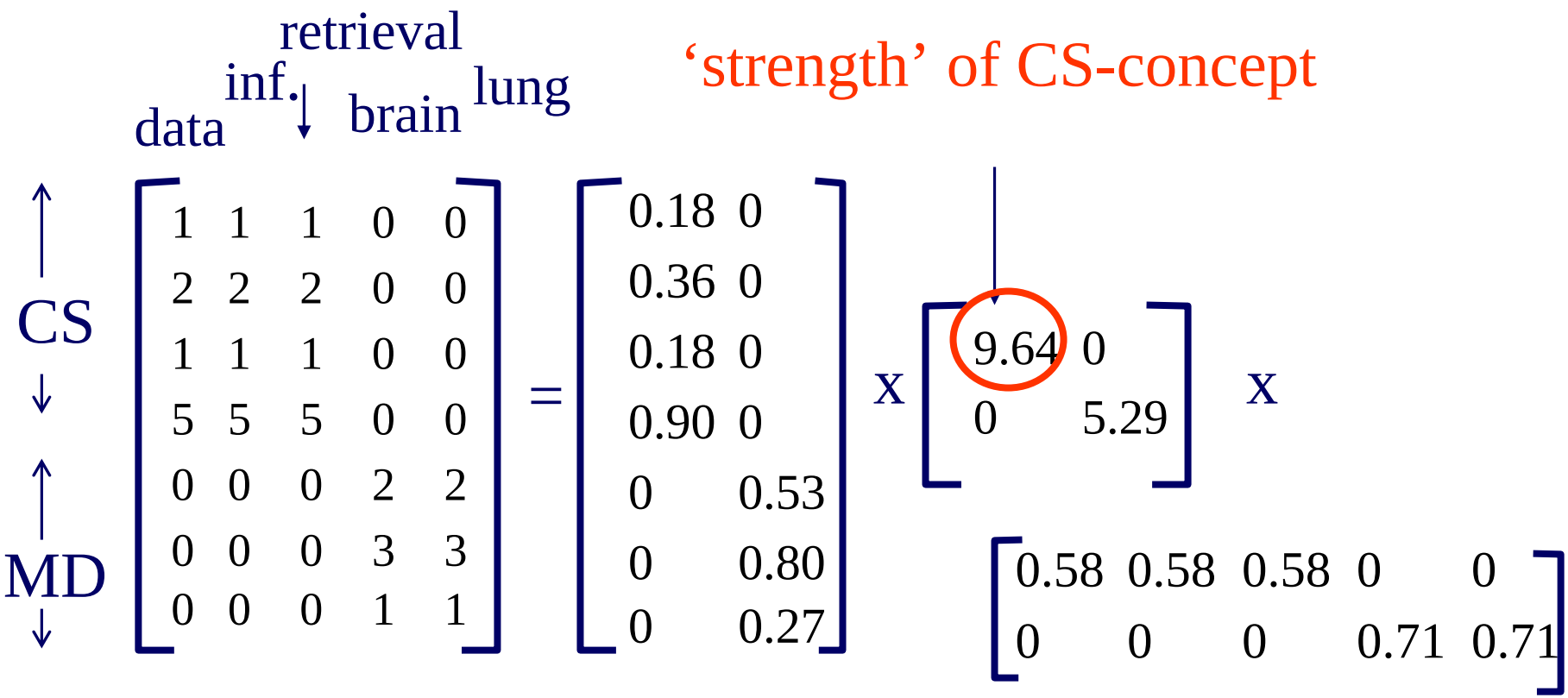
- **A = U Σ $V^T$** - example:

doc-to-concept similarity matrix



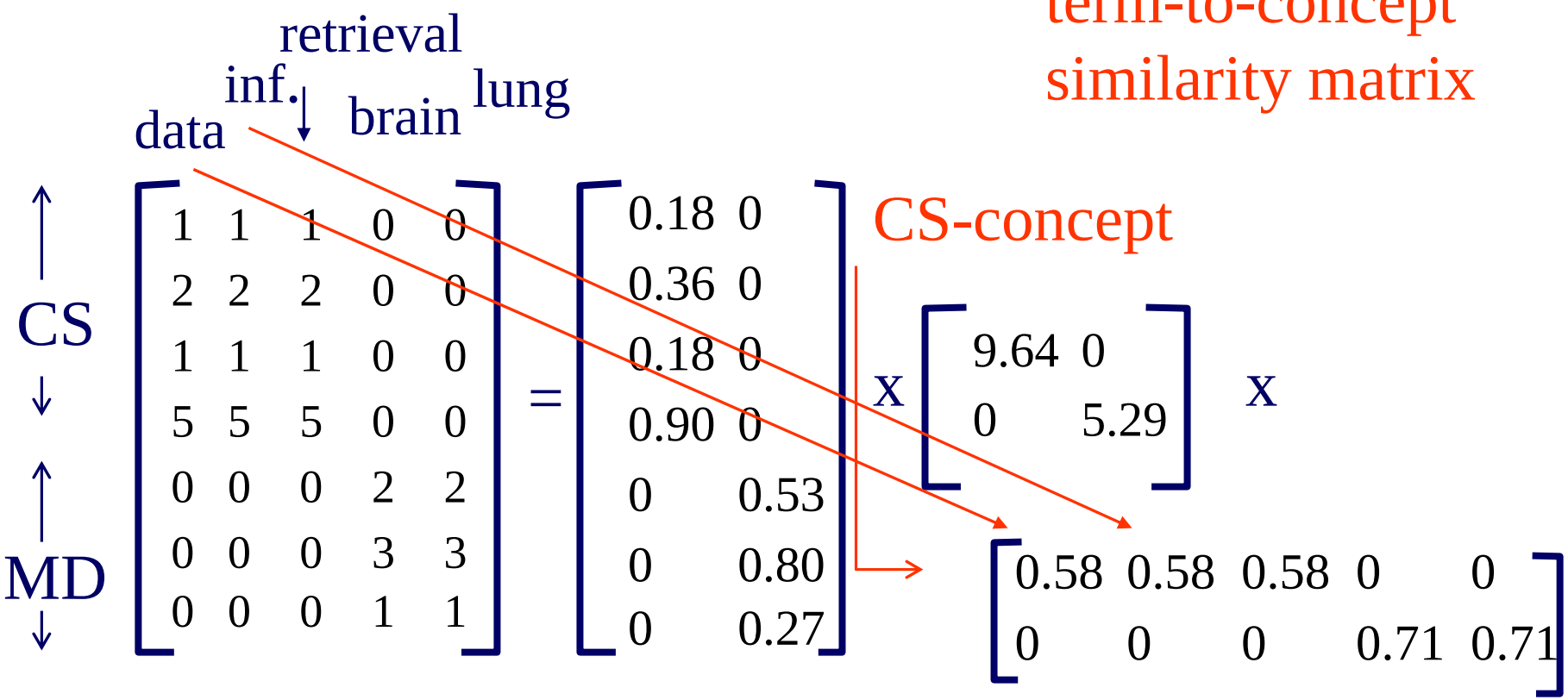
# SVD - Example

- **A = U Σ $V^T$** - example:



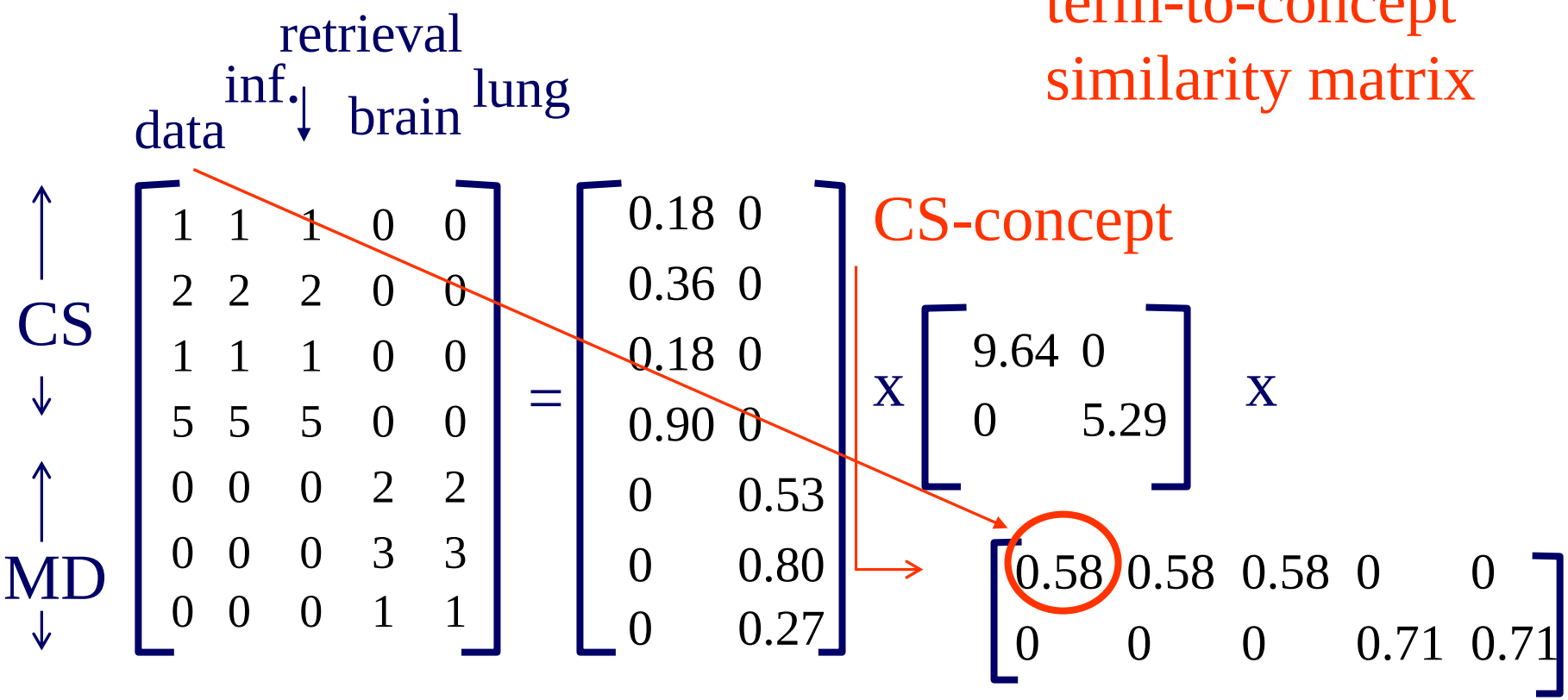
# SVD - Example

- **A = U Σ $V^T$** - example:



# SVD - Example

- **A = U Σ $V^T$** - example:



# SVD - Interpretation

'documents', 'terms' and 'concepts':

Q: if **A** is the document-to-term matrix, what is $\mathbf{A}^T \mathbf{A}$?

A: term-to-term ([m x m]) similarity matrix

Q: $\mathbf{A}\ \mathbf{A}^T$ ?

A: document-to-document ([n x n]) similarity matrix

# SVD properties

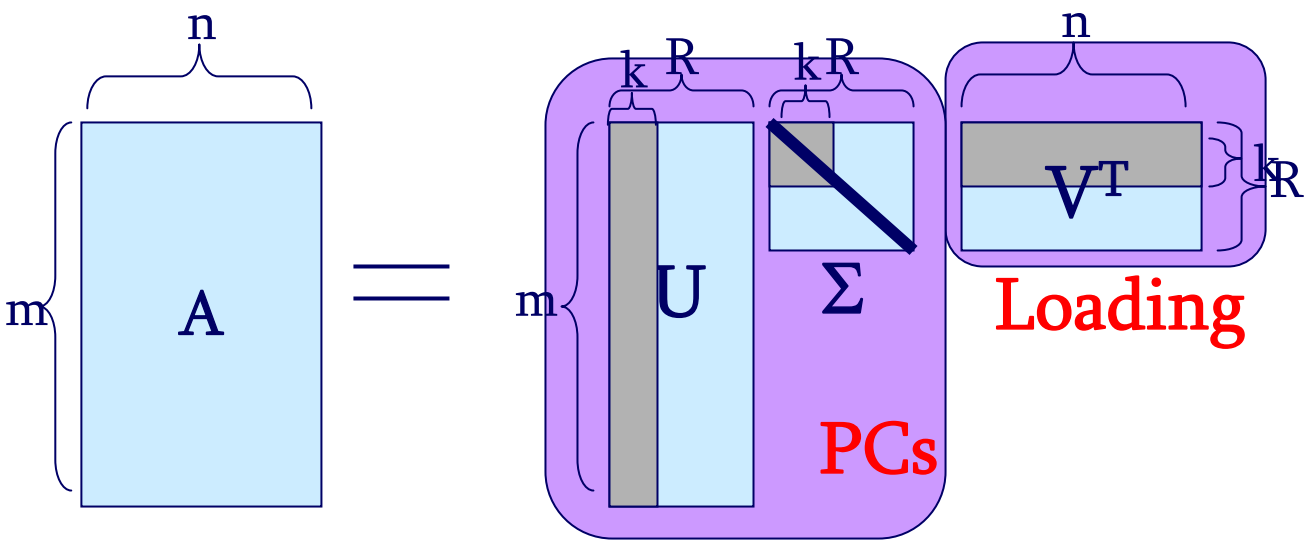
- **V** are the eigenvectors of the *covariance matrix* $\mathbf{A}^T\mathbf{A}$

- **U** are the eigenvectors of the *Gram (inner-product) matrix* $\mathbf{A}\mathbf{A}^T$

Further reading:
1. Ian T. Jolliffe, *Principal Component Analysis* (2nd ed), Springer, 2002.
2. Gilbert Strang, *Linear Algebra and Its Applications* (4th ed), Brooks Cole, 2005.

# Principal Component Analysis (PCA)

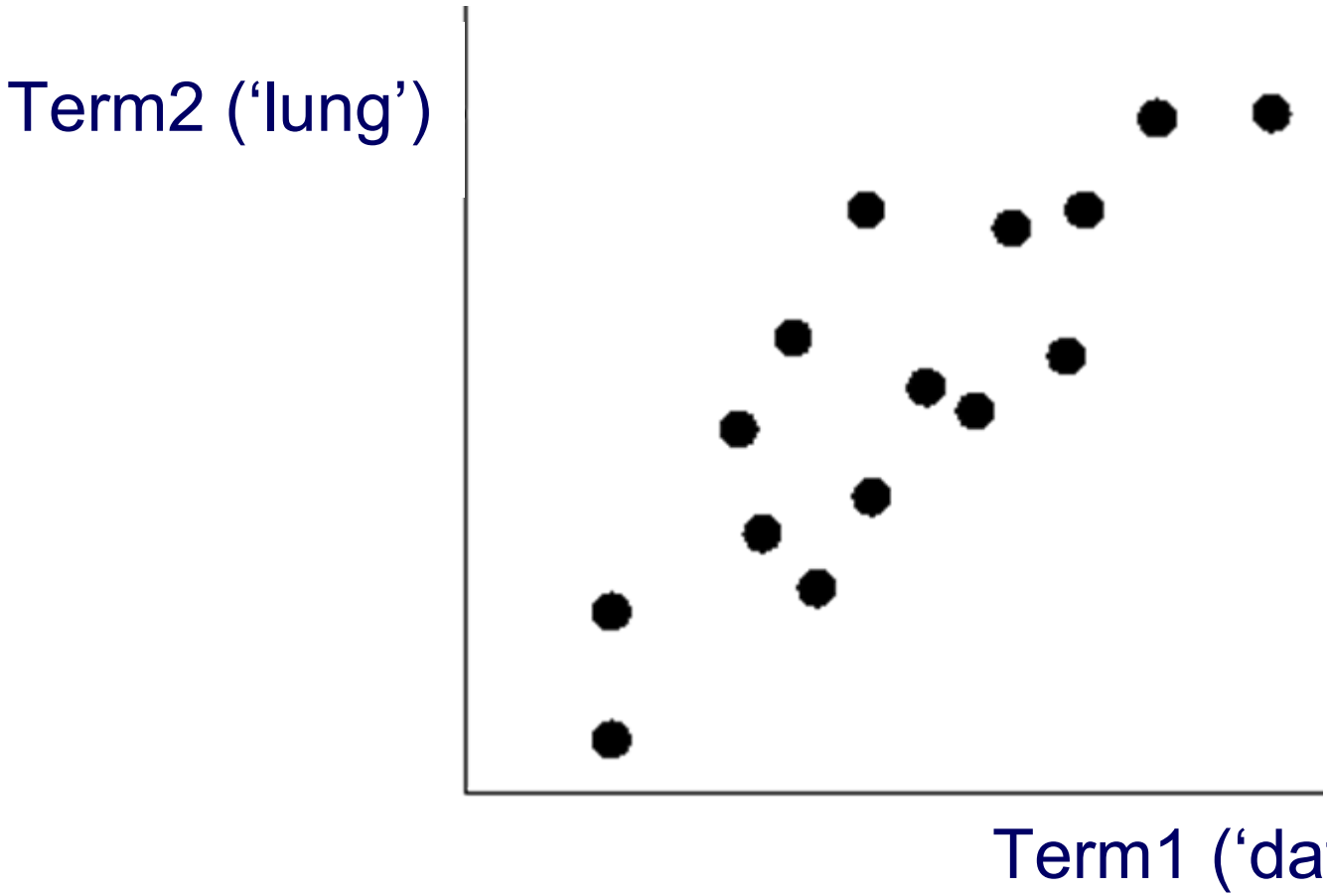
- SVD $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$



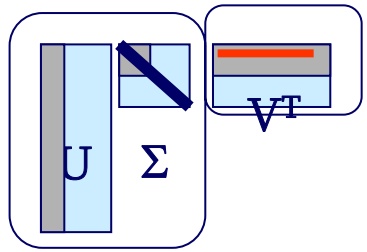

  - PCA is an important application of SVD
  - Note that U and V are dense and may have negative entries

# PCA interpretation

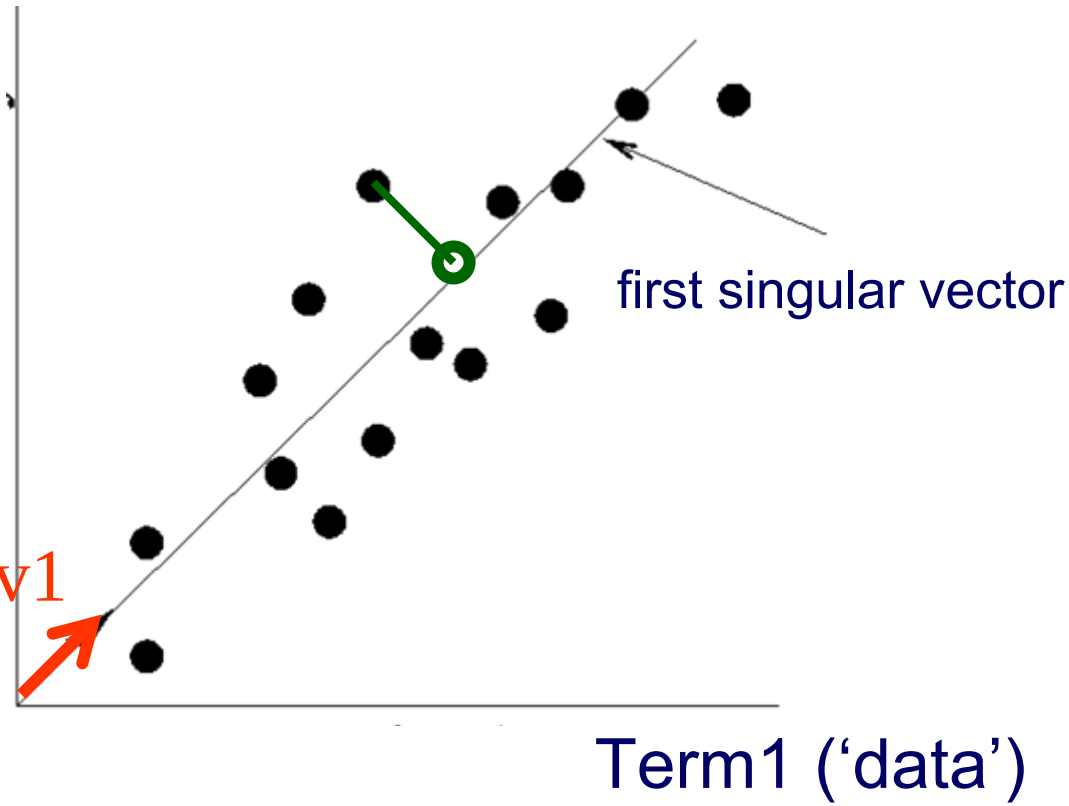
- best axis to project on: ('best' = min sum of squares of projection errors)



# PCA - interpretation




PCA projects points
Onto the “best” axis




- minimum RMS error

# Roadmap

- Motivation
- Matrix tools
- Tensor tools
- Case studies

- SVD, PCA
- **HITS, PageRank**
- CUR
- Co-clustering

# Kleinberg's algorithm HITS

- Problem dfn: given the web and a query
- find the most 'authoritative' web pages for this query

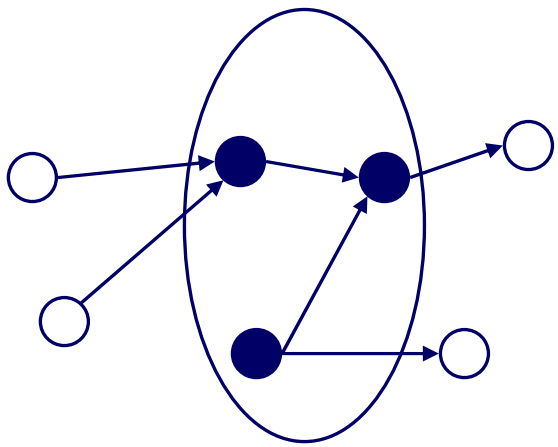
Step 0: find all pages containing the query terms

Step 1: expand by one move forward and backward

Further reading:
1. J. Kleinberg. Authoritative sources in a hyperlinked environment. SODA 1998

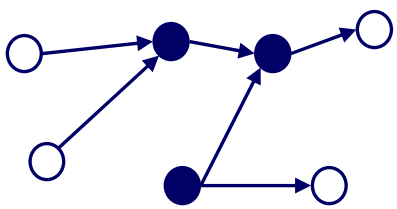
# Kleinberg's algorithm HITS

- Step 1: expand by one move forward and backward

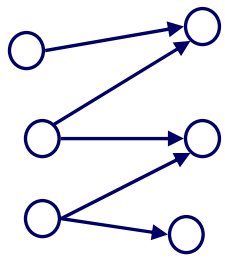


# Kleinberg's algorithm HITS

- on the resulting graph, give high score (= 'authorities') to nodes that many important nodes point to
- give high importance score ('hubs') to nodes that point to good 'authorities'



hubs



authorities

# Kleinberg's algorithm HITS

observations

- recursive definition!
- each node (say, '*i*'-th node) has both an authoritativeness score $a_i$ and a hubness score $h_i$

# Kleinberg's algorithm: HITS

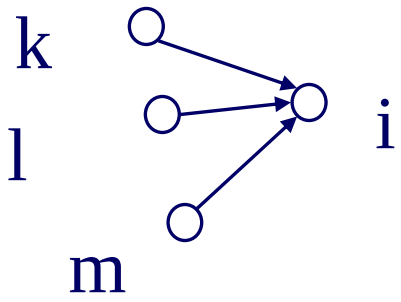
Let **A** be the adjacency matrix:

the $(i,j)$ entry is 1 if the edge from $i$ to $j$ exists

Let **h** and **a** be [n x 1] vectors with the 'hubness' and 'authoritativiness' scores.

Then:

# Kleinberg's algorithm: HITS




Then:

$$a_i = h_k + h_l + h_m$$

that is

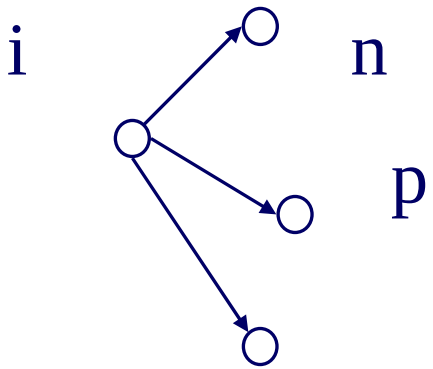
$a_i$ = Sum ($h_j$) over all $j$ that ($j$,$i$) edge exists

or

$$\mathbf{a} = \mathbf{A}^T \mathbf{h}$$

# Kleinberg's algorithm: HITS

symmetrically, for the 'hubness':




$$h_i = a_n + a_p + a_q$$

that is

$h_i$ = Sum ($q_j$) over all $j$ that ($i$,$j$) edge exists

or

**h = A a**

# Kleinberg's algorithm: HITS

In conclusion, we want vectors **h** and **a** such that:

$$\mathbf{h} = \mathbf{A}\,\mathbf{a}$$

$$\mathbf{a} = \mathbf{A}^T\,\mathbf{h}$$

That is:

$$\mathbf{a} = \mathbf{A}^T\mathbf{A}\,\mathbf{a}$$

# Kleinberg's algorithm: HITS

**a** is a right singular vector of the adjacency matrix **A** (by dfn!), a.k.a the eigenvector of $\mathbf{A}^T\mathbf{A}$

Starting from random **a'** and iterating, we'll eventually converge

Q: to which of all the eigenvectors? why?

A: to the one of the strongest eigenvalue,

$$(\mathbf{A}^T\mathbf{A})^k\,\mathbf{a} = \lambda_1^k\,\mathbf{a}$$

# Kleinberg's algorithm - discussion

- 'authority' score can be used to find 'similar pages' (how?)
- closely related to 'citation analysis', social networks / 'small world' phenomena

See also **TOPHITS**

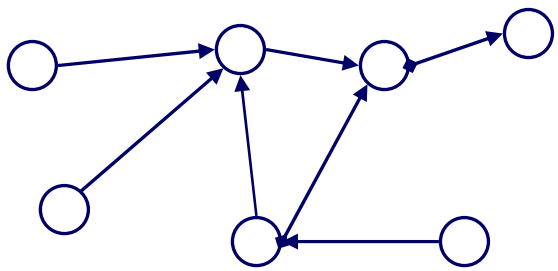
# Roadmap

- Motivation
- Matrix tools
- Tensor tools
- Case studies

- SVD, PCA
- HITS, **PageRank**
- CUR
- Co-clustering

# Motivating problem: PageRank

Given a directed graph, find its most interesting/central node

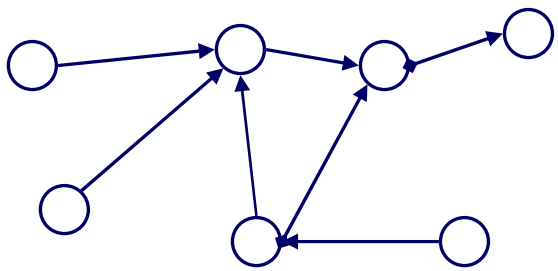


A node is important, if it is connected with important nodes (recursive, but OK!)

# Motivating problem – PageRank solution

Given a directed graph, find its most interesting/central node

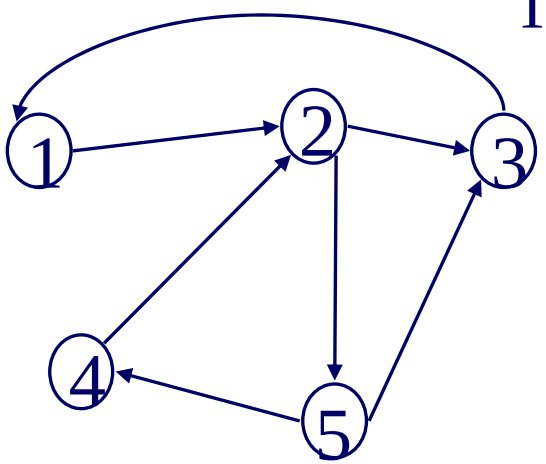
Proposed solution: Random walk; spot most 'popular' node (-> steady state prob. (ssp))



A node has high ssp, if it is connected with high ssp nodes (recursive, but OK!)

# (Simplified) PageRank algorithm

- Let **A** be the transition matrix (= adjacency matrix); let **B** be the transpose, column-normalized - then

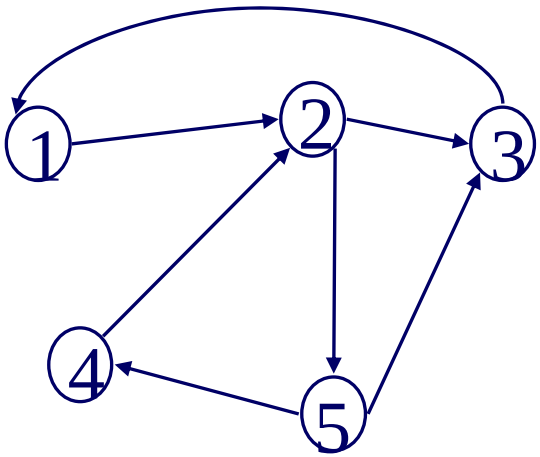



From / To

$$\mathbf{B} \begin{bmatrix} & & 1 & & \\ 1 & & & 1 & \\ & 1/2 & & & 1/2 \\ & & & & 1/2 \\ & 1/2 & & & \end{bmatrix} \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix} = \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix}$$

# (Simplified) PageRank algorithm

- **B p = p**




**B** **p** = **p**

| | | | | |
|---|---|---|---|---|
| | | 1 | | |
| 1 | | | 1 | |
| | 1/2 | | | 1/2 |
| | | | | 1/2 |
| | 1/2 | | | |

| |
|---|
| p1 |
| p2 |
| p3 |
| p4 |
| p5 |

=

| |
|---|
| p1 |
| p2 |
| p3 |
| p4 |
| p5 |

# (Simplified) PageRank algorithm

- **B p** = 1 * **p**
- thus, **p** is the **eigenvector** that corresponds to the highest eigenvalue (=1, since the matrix is column-normalized)
- Why does such a **p** exist?
  - **p** exists if **B** is nxn, nonnegative, irreducible [Perron–Frobenius theorem]

# (Simplified) PageRank algorithm

- In short: imagine a particle randomly moving along the edges
- compute its steady-state probabilities (ssp)

Full version of algo: with occasional random jumps

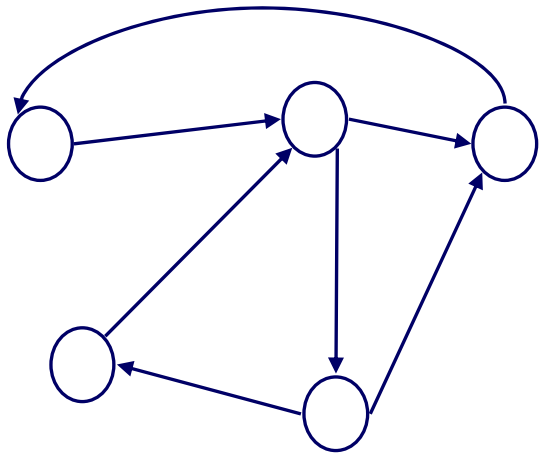
Why? To make the matrix irreducible

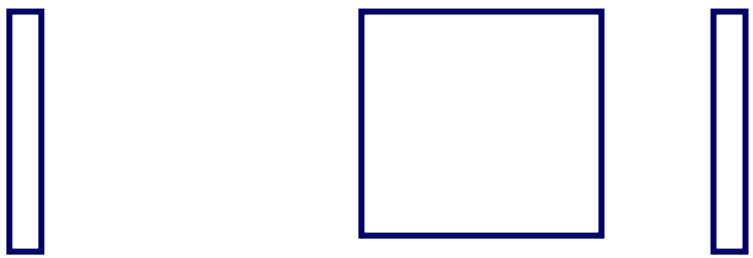
# Full Algorithm

- With probability *1-c*, fly-out to a random node
- Then, we have

$$\mathbf{p} = c\,\mathbf{B}\,\mathbf{p} + (1-c)/n\,\mathbf{1} \Rightarrow$$

$$\mathbf{p} = (1-c)/n\,[\mathbf{I} - c\,\mathbf{B}]^{-1}\,\mathbf{1}$$





# Roadmap

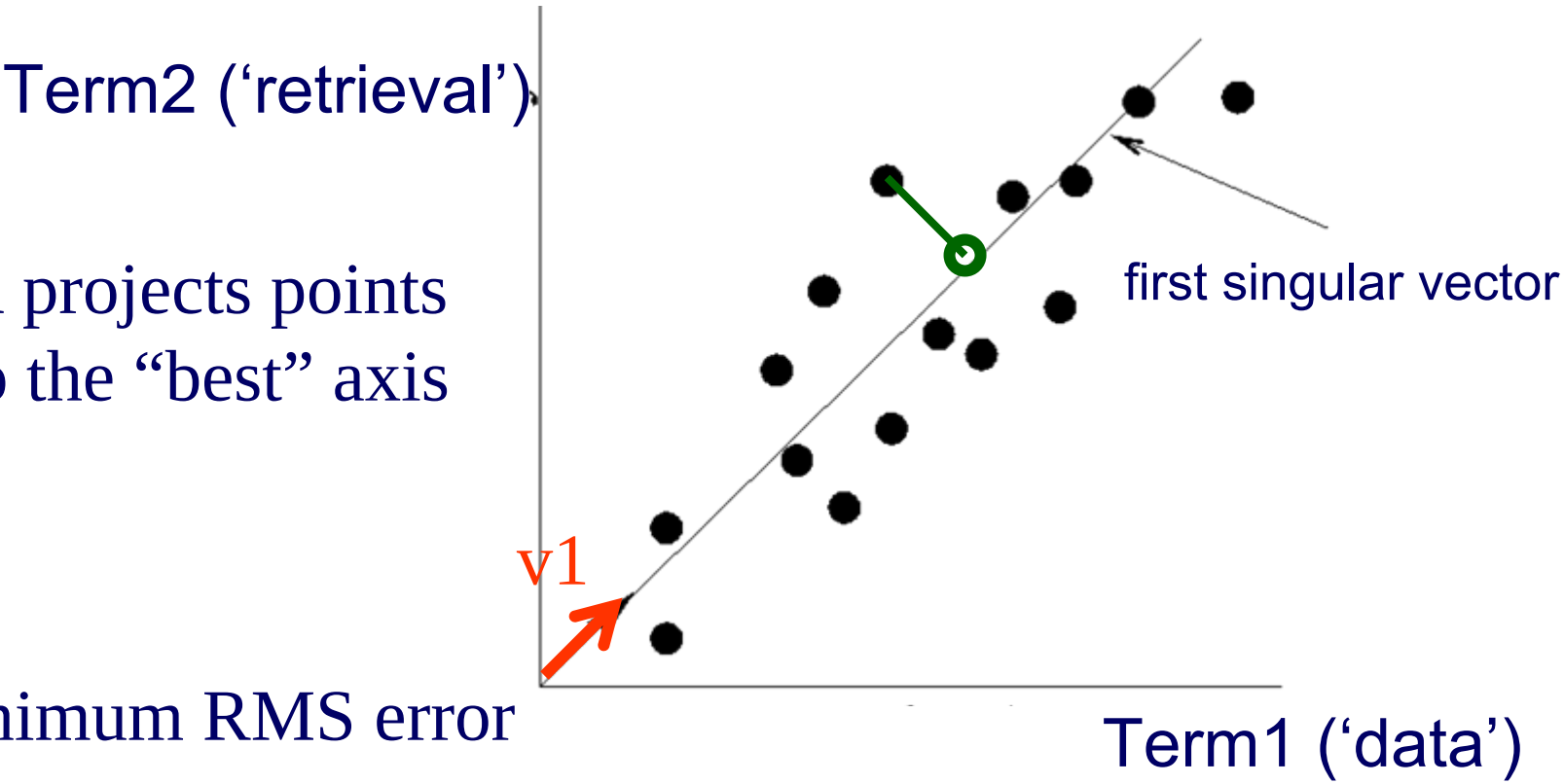
- Motivation
- Matrix tools
- Tensor tools
- Case studies

- SVD, PCA
- HITS, PageRank
- **CUR**
- Co-clustering

# Motivation of CUR or CMD

- SVD, PCA all transform data into some abstract space (specified by a set basis)
  - Interpretability problem
  - Loss of sparsity

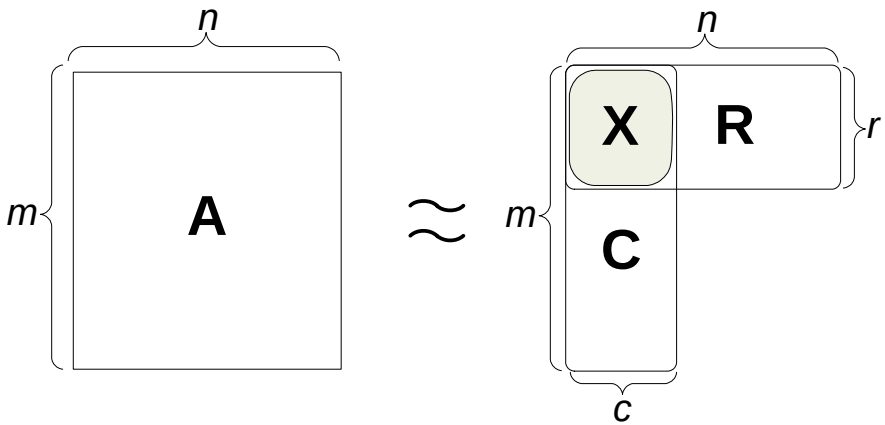
# PCA - interpretation



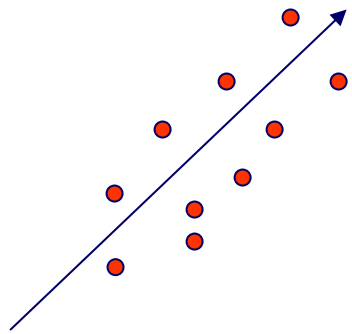

- minimum RMS error

# CUR

- Example-based projection: use actual rows and columns to specify the subspace
- Given a matrix $A \in R^{m \times n}$, find three matrices $C \in R^{m \times c}$, $U \in R^{c \times r}$, $R \in R^{r \times n}$, such that $||A-CUR||$ is small



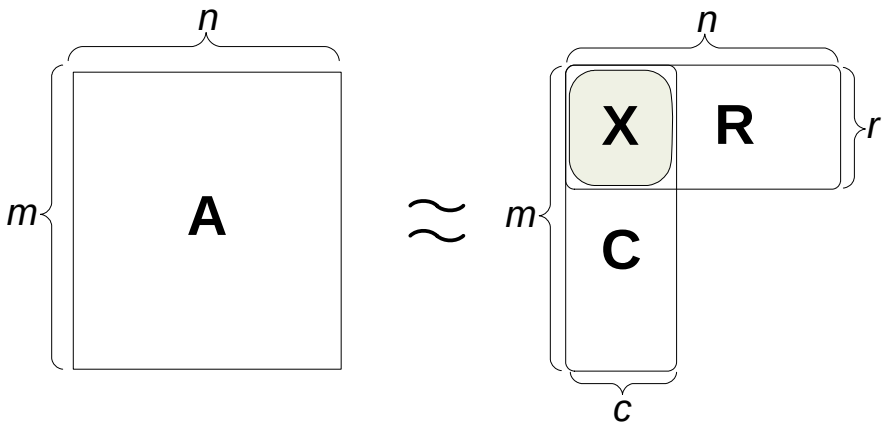

U is the pseudo-inverse of X

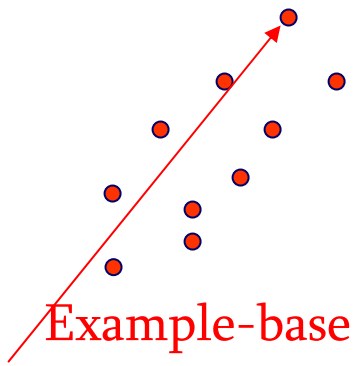


Orthogonal projection

# CUR

- Example-based projection: use actual rows and columns to specify the subspace
- Given a matrix $A \in R^{m \times n}$, find three matrices $C \in R^{m \times c}$, $U \in R^{c \times r}$, $R \in R^{r \times n}$, such that $||A-CUR||$ is small




U is the pseudo-inverse of X:

$U = X^{\dagger} = (U^T U)^{-1} U^T$

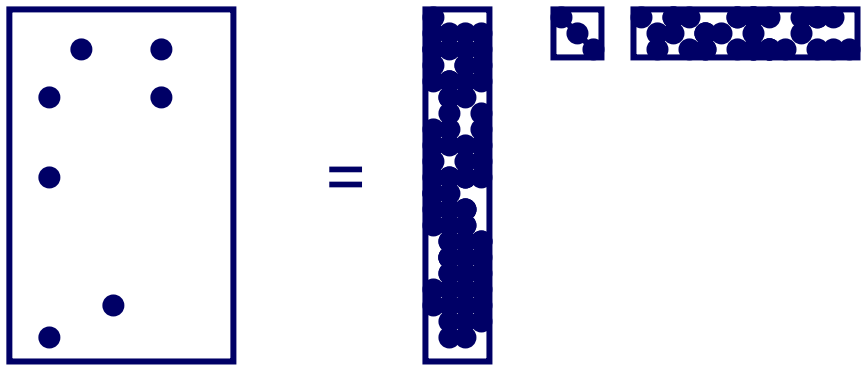
# CUR (cont.)

- Key question:
  - How to select/sample the columns and rows?
- Uniform sampling
- Biased sampling
  - CUR w/ absolute error bound
  - CUR w/ relative error bound

Reference:

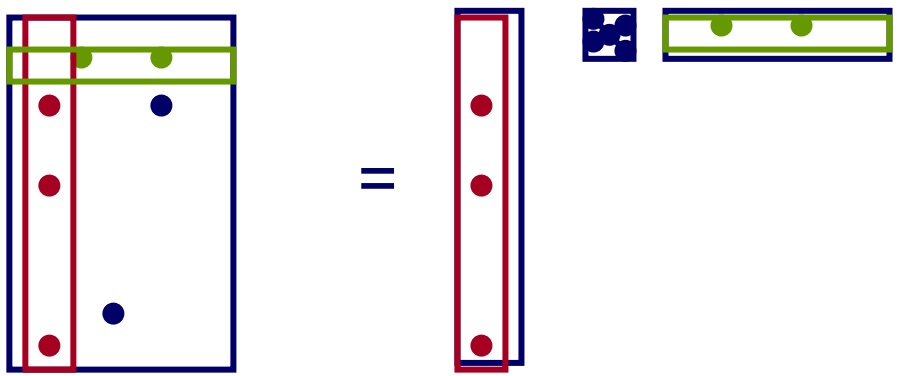
1. Tutorial: Randomized Algorithms for Matrices and Massive Datasets, SDM'06
2. Drineas et al. Subspace Sampling and Relative-error Matrix Approximation: Column-Row-Based Methods, ESA2006
3. Drineas et al., Fast Monte Carlo Algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition, SIAM Journal on Computing, 2006.

# The sparsity property – pictorially:



SVD/PCA:
Destroys sparsity

U Σ $V^T$



CUR: maintains sparsity

C U R

# The sparsity property

sparse and small

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

Big but sparse

Big and dense

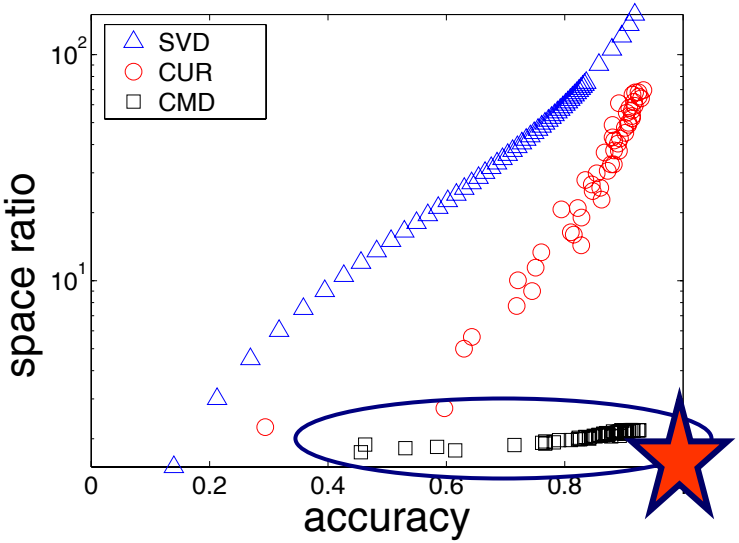
dense but small

CUR: $A = C U R$

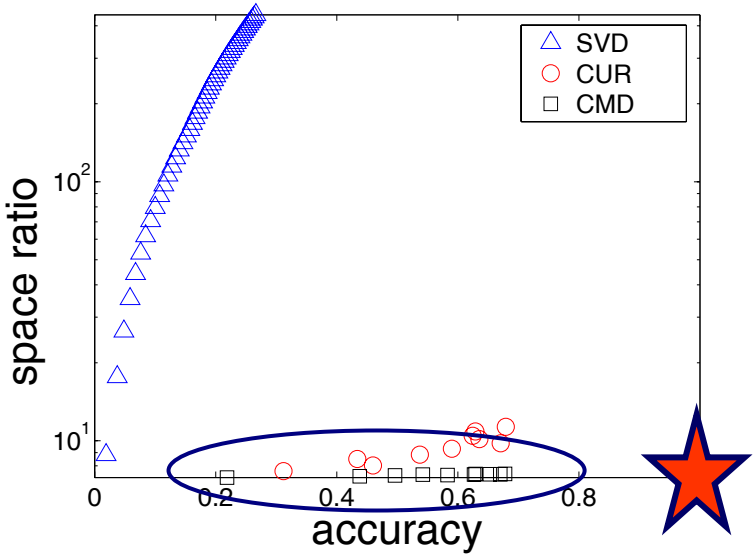
Big but sparse

Big but sparse

# The sparsity property (cont.)




Network




DBLP

- CMD uses much smaller space to achieve the same accuracy
- CUR limitation: duplicate columns and rows
- SVD limitation: orthogonal projection densifies the data

Reference:
Sun et al. Less is More: Compact Matrix Decomposition for Large Sparse Graphs, SDM'07

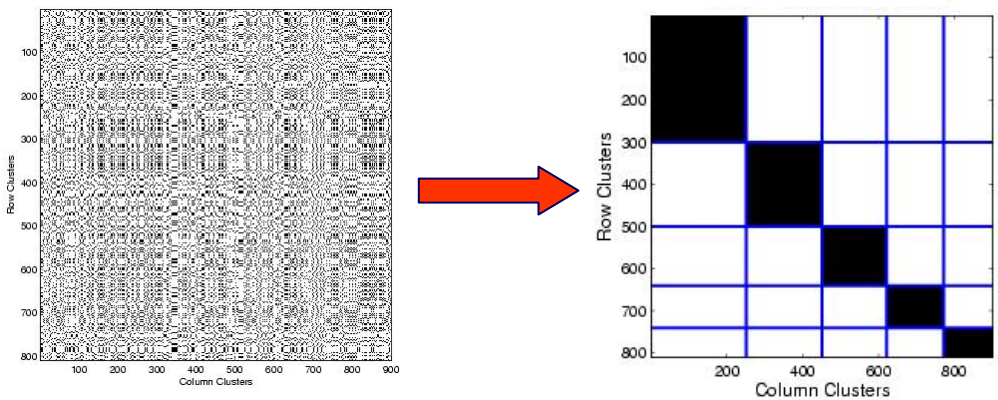
# Roadmap

- Motivation
- Matrix tools
- Tensor tools
- Case studies

- SVD, PCA
- HITS, PageRank
- CUR
- **Co-clustering**

# Co-clustering

- Given data matrix and the number of row and column groups *k* and *l*
- Simultaneously
  - Cluster rows of *p(X, Y)* into *k* disjoint groups
  - Cluster columns of *p(X, Y)* into *l* disjoint groups



# Co-clustering

- Let $X$ and $Y$ be discrete random variables
  - $X$ and $Y$ take values in $\{1, 2, \ldots, m\}$ and $\{1, 2, \ldots, n\}$
  - $p(X, Y)$ denotes the joint probability distribution—if not known, it is often estimated based on co-occurrence data
  - Application areas: text mining, market-basket analysis, analysis of browsing behavior, etc.
- Key Obstacles in Clustering Contingency Tables
  - High Dimensionality, Sparsity, Noise
  - Need for robust and scalable algorithms

Reference:
1. Dhillon et al. Information-Theoretic Co-clustering, KDD'03

$$\underset{m}{\begin{bmatrix} .05 & .05 & .05 & 0 & 0 & 0 \\ .05 & .05 & .05 & 0 & 0 & 0 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ .04 & .04 & 0 & .04 & .04 & .04 \\ .04 & .04 & .04 & 0 & .04 & .04 \end{bmatrix}}$$

$n$

eg, terms x documents

$$\underset{m}{\begin{bmatrix} .5 & 0 & 0 \\ .5 & 0 & 0 \\ 0 & .5 & 0 \\ 0 & .5 & 0 \\ 0 & 0 & .5 \\ 0 & 0 & .5 \end{bmatrix}} \quad \underset{k}{\begin{bmatrix} .3 & 0 \\ 0 & .3 \\ .2 & .2 \end{bmatrix}} \quad \underset{l}{\begin{bmatrix} .36 & .36 & .28 & 0 & 0 & 0 \\ 0 & 0 & 0 & .28 & .36 & .36 \end{bmatrix}} = \begin{bmatrix} .054 & .054 & .042 & 0 & 0 & 0 \\ .054 & .054 & .042 & 0 & 0 & 0 \\ 0 & 0 & 0 & .042 & .054 & .054 \\ 0 & 0 & 0 & .042 & .054 & .054 \\ .036 & .036 & 028 & .028 & .036 & .036 \\ .036 & .036 & .028 & .028 & .036 & .036 \end{bmatrix}$$

$k$ $l$ $n$

med. doc

cs doc

$$\begin{bmatrix} .05 & .05 & .05 & 0 & 0 & 0 \\ .05 & .05 & .05 & 0 & 0 & 0 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ .04 & .04 & 0 & .04 & .04 & .04 \\ .04 & .04 & .04 & 0 & .04 & .04 \end{bmatrix}$$

med. terms

cs terms

common terms

term group x doc. group

$$\begin{bmatrix} .5 & 0 & 0 \\ .5 & 0 & 0 \\ 0 & .5 & 0 \\ 0 & .5 & 0 \\ 0 & 0 & .5 \\ 0 & 0 & .5 \end{bmatrix} \begin{bmatrix} .3 & 0 \\ 0 & .3 \\ .2 & .2 \end{bmatrix} \begin{bmatrix} .36 & .36 & .28 & 0 & 0 & 0 \\ 0 & 0 & 0 & .28 & .36 & .36 \end{bmatrix} =$$

$$\left[\begin{array}{ccc|ccc} .054 & .054 & .042 & 0 & 0 & 0 \\ .054 & .054 & .042 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & .042 & .054 & .054 \\ 0 & 0 & 0 & .042 & .054 & .054 \\ \hline .036 & .036 & 028 & .028 & .036 & .036 \\ .036 & .036 & .028 & .028 & .036 & .036 \end{array}\right]$$

doc x doc group

term x term-group

# Co-clustering

Observations

- uses KL divergence, instead of L2
- the middle matrix is **not** diagonal
  - we'll see that again in the Tucker tensor decomposition

# Problem with Information Theoretic Co-clustering

- Number of row and column groups must be specified

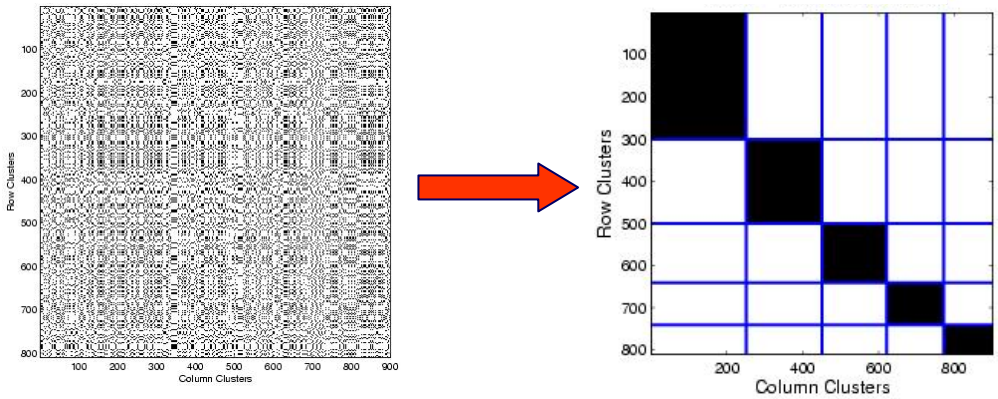
Desiderata:

✓ Simultaneously discover row and column groups

✘ Fully Automatic: No “magic numbers”

✓ Scalable to large graphs

# Cross-association




Desiderata:

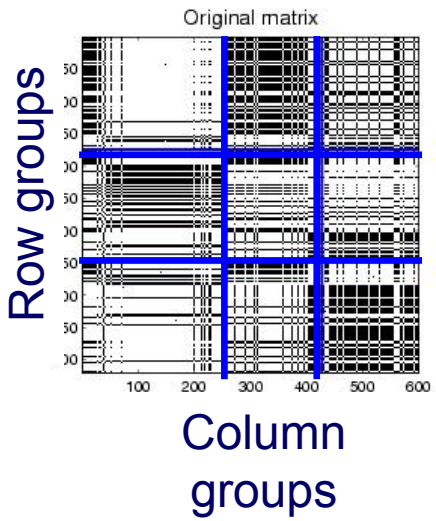
- ✓ Simultaneously discover row and column groups
- ✓ Fully Automatic: No “magic numbers”
- ✓ Scalable to large matrices

Reference:

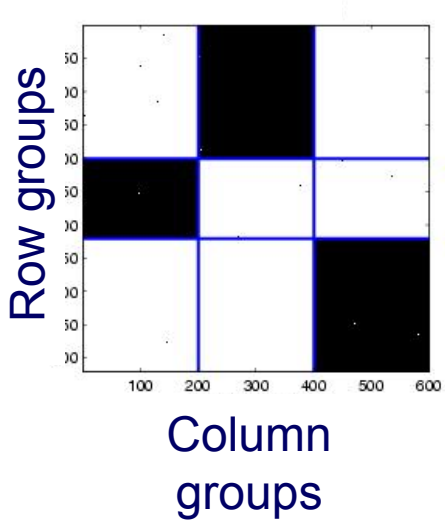
1. Chakrabarti et al. Fully Automatic Cross-Associations, KDD’04

Sandia National Laboratories

# What makes a cross-association "good"?



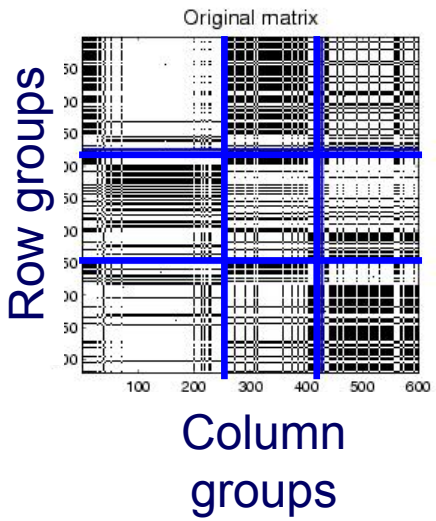

versus



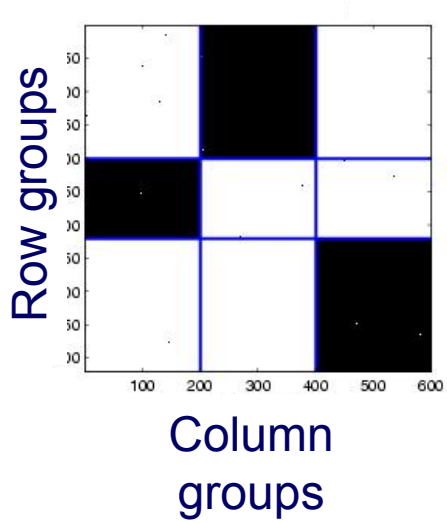

**Why is this better?**

Sandia National Laboratories

# What makes a cross-association "good"?




versus

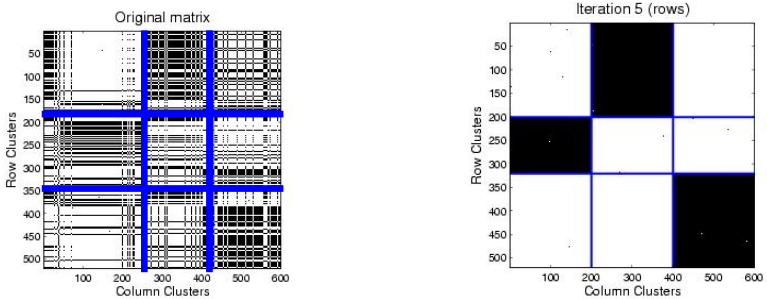



**Why is this better?**

simpler; easier to describe

**easier to compress**!

# What makes a cross-association "good"?







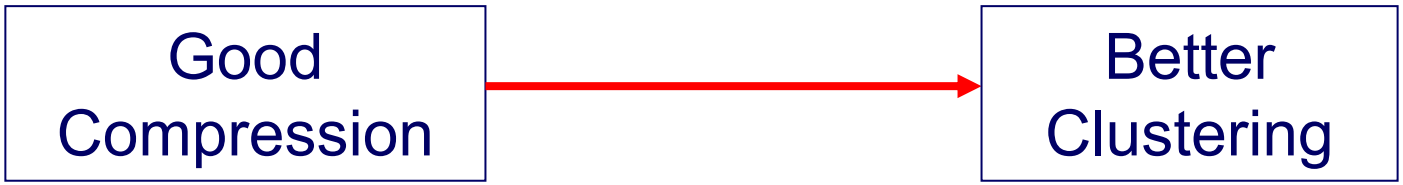
Problem definition: given an encoding scheme

- decide on the # of col. and row groups *k* and *l*
- and reorder rows and columns,
- to achieve best compression

details

Sandia National Laboratories

# Main Idea




Total Encoding Cost = $\Sigma_i$ size$_i$ * H(x$_i$) + Cost of describing cross-associations

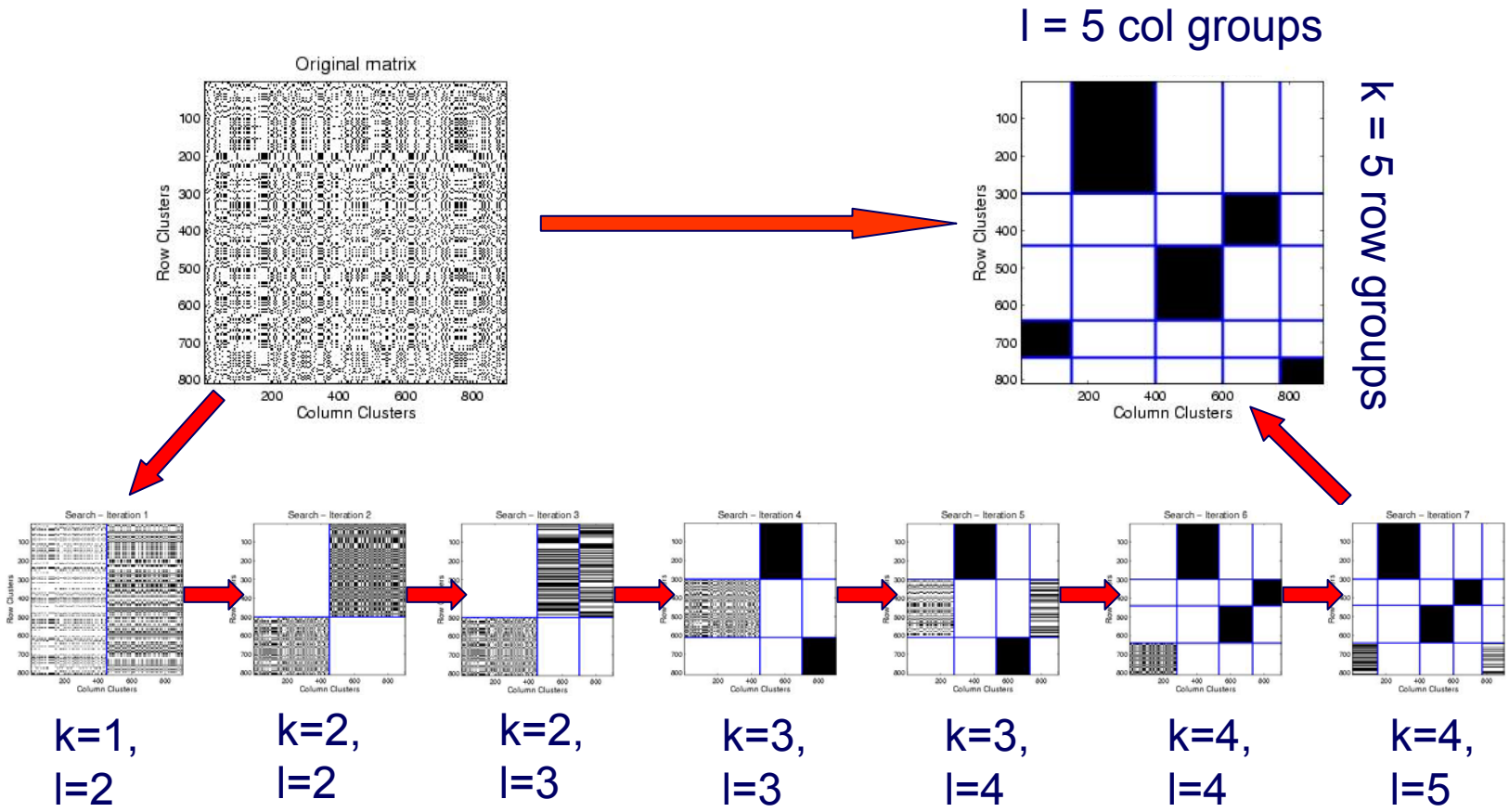
Code Cost

Description Cost

Minimize the total cost (# bits)

for lossless compression

# Algorithm


l = 5 col groups
k = 5 row groups
Original matrix
Row Clusters
Column Clusters
Search – Iteration 1
Search – Iteration 2
Search – Iteration 3
Search – Iteration 4
Search – Iteration 5
Search – Iteration 6
Search – Iteration 7
k=1, l=2
k=2, l=2
k=2, l=3
k=3, l=3
k=3, l=4
k=4, l=4
k=4, l=5

# Algorithm

Code for cross-associations (matlab):

www.cs.cmu.edu/~deepay/mywww/software/CrossAssociations-01-27-2005.tgz

Variations and extensions:

- 'Autopart' [Chakrabarti, PKDD'04]
- www.cs.cmu.edu/~deepay

# Matrix tools - summary

- SVD:
  - optimal for L2 – VERY popular (HITS, PageRank, Karhunen-Loeve, Latent Semantic Indexing, PCA, etc etc)
- C-U-R (CMD etc)
  - near-optimal; sparsity; interpretability