Carnegie Mellon University
School of Computer Science

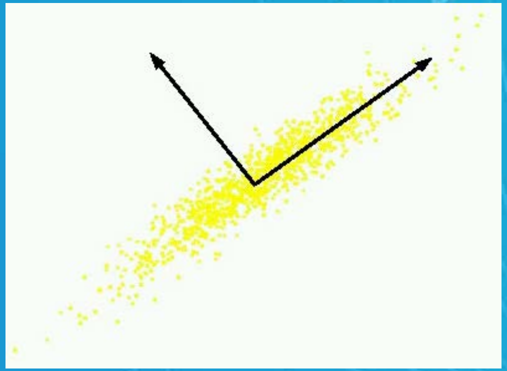
10701: Introduction to Machine Learning

Dimensionality Reduction and Sub-Space Analysis:

PCA, SVD, Manifold, and beyond

Eric Xing

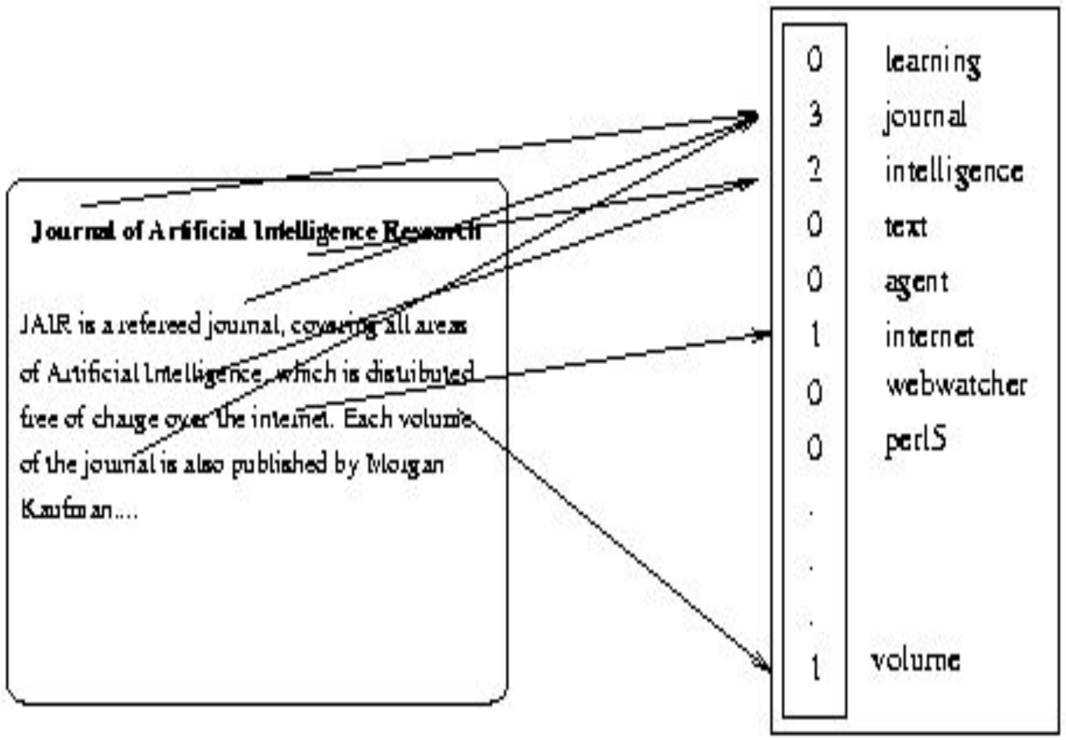
Lecture 15, October 26, 2020



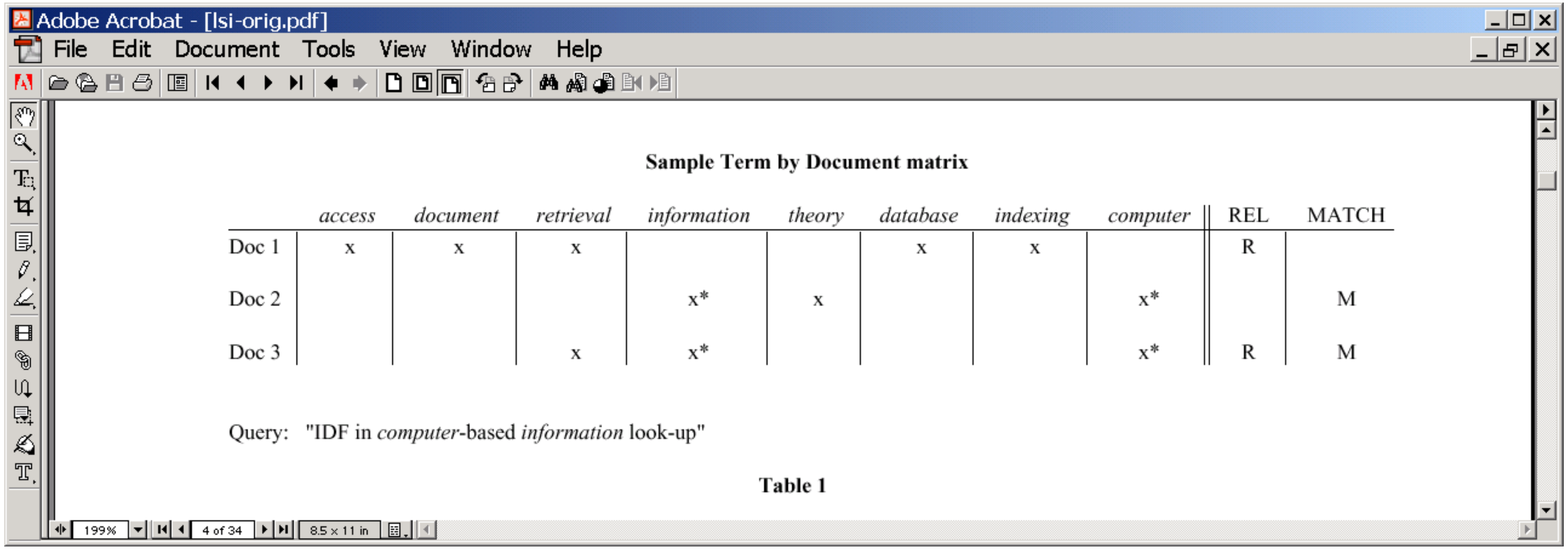
Reading: Chap 12.1, CB book

Text document retrieval/labelling

- Represent each document by a high-dimensional vector in the space of words



Example



Sample Term by Document matrix

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

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

Table 1

-- Relevant docs may not have the query terms

→ but may have many "related" terms

-- Irrelevant docs may *have* the query terms

→ but may not have any "related" terms

Problems

- Looks for literal term matches
  - Terms in queries (esp short ones) don't always capture user's information need well
- Problems:
  - Synonymy: other words with the same meaning
    - Car and automobile
  - No associations between words are made in the vector space representation.

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

  - Polysemy: the same word having other meanings
    - Apple (fruit and company)
  - The vector space model is unable to discriminate between different meanings of the same word.

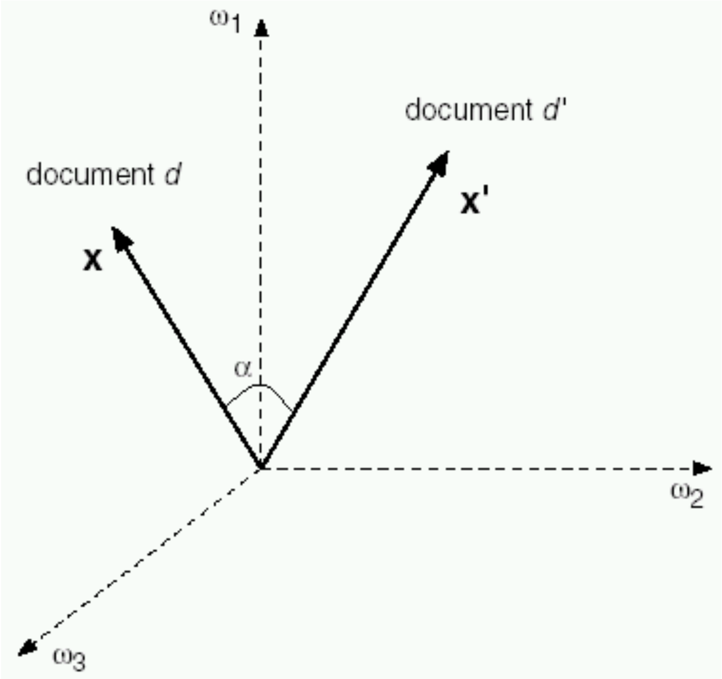
$$\mathrm{sim}_{\mathtt{true}}(d, q) < \cos(\angle(\vec{d}, \vec{q}))$$

- What if we could match against 'concepts', that represent related words, rather than words themselves

The task:

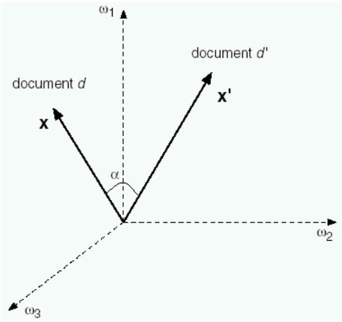
- Say, we want to have a mapping …, so that

$\Rightarrow$



- Compare similarity
- Classify contents
- Cluster/group/categorize docs
- Distill semantics and perspectives
- ..

Latent Semantic Indexing (LSI) (Deerwester et al., 1990)



- Uses statistically derived conceptual indices instead of individual words for retrieval
- Assumes that there is some underlying or *latent* structure in word usage that is obscured by variability in word choice
- Key idea: instead of representing documents and queries as vectors in a t-dim space of terms
  - Represent them (and terms themselves) as vectors in a lower-dimensional space whose axes are concepts that effectively group together similar words
  - Uses SVD (and now many other methods) to reduce document representations,
  - The axes are the Principal Components (or topics, basis, …) from such analysis

More General Motivations: Factor or Component Analysis

- We study phenomena that can not be directly observed
  - ego, personality, intelligence in psychology
  - Underlying factors that govern the observed data

- We want to identify and operate with underlying latent factors rather than the observed data
  - E.g. topics in news articles
  - Transcription factors in genomics

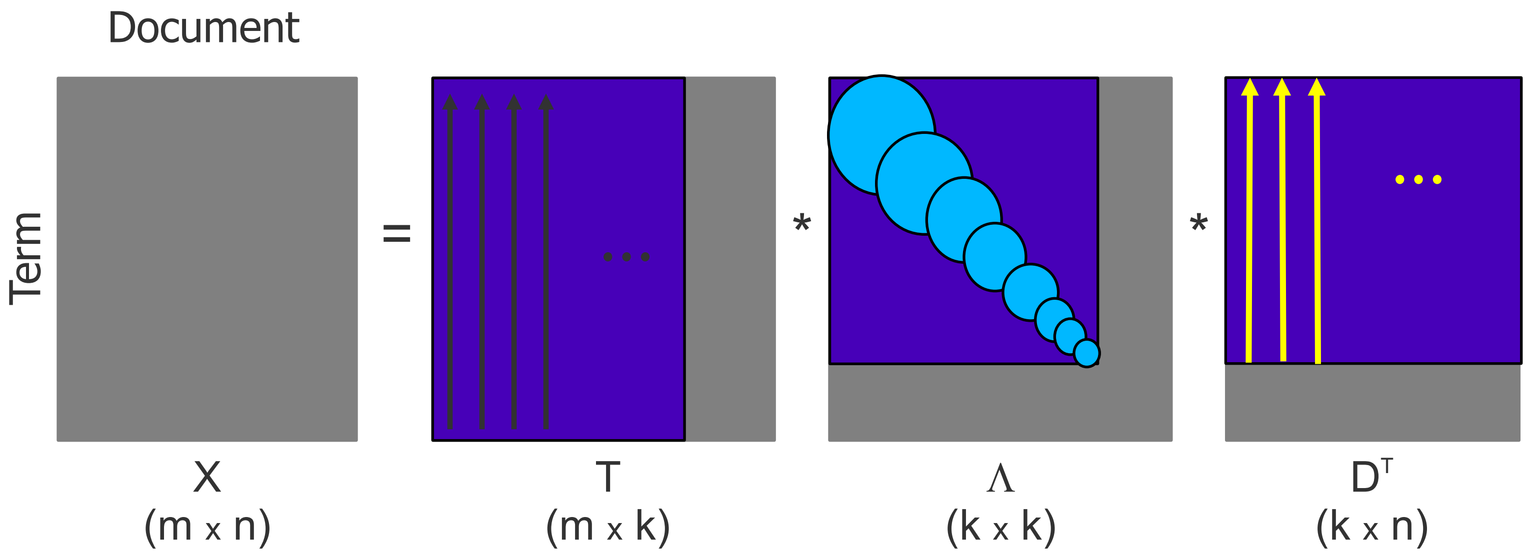
- We want to discover and exploit hidden relationships
  - “beautiful car” and “gorgeous automobile” are closely related
  - So are “driver” and “automobile”
  - But does your search engine know this?
  - Reduces noise and error in results

More General Motivations: Dimensionality Reduction

- We have too many observations and dimensions
  - To reason about or obtain insights from
  - To visualize
  - Too much noise in the data
  - Need to “reduce” them to a smaller set of factors
  - Better representation of data without losing much information
  - Can build more effective data analyses on the reduced-dimensional space: classification, clustering, pattern recognition

- Combinations of observed variables may be more effective bases for insights, even if the physical meaning of these “synthetic” entities is obscure

Subspace analysis



cluster/topic/basis Distributions (subspace)

***A priori* weights**

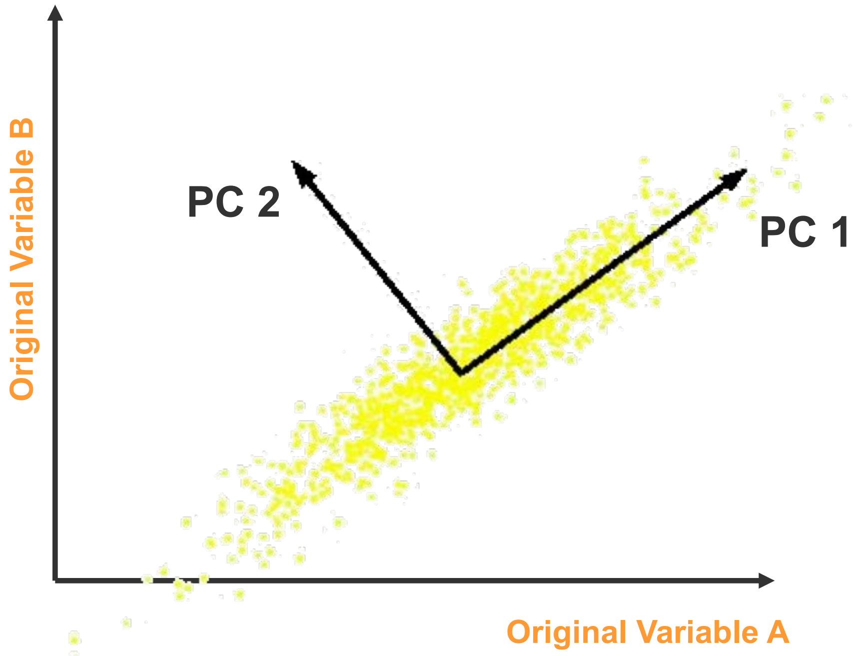
Memberships (coordinates)

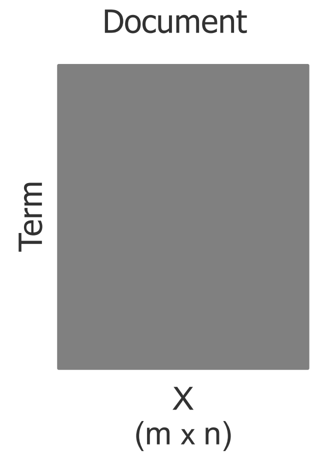
- Clustering: (0,1) matrix
- LSI/NMF: "arbitrary" matrices
- Topic Models: stochastic matrix
- Sparse coding: "arbitrary" **sparse** matrices

Basic Concept

- Areas of variance in data are where items can be best discriminated and key underlying phenomena observed
- If two items or dimensions are highly correlated or dependent
  - They are likely to represent highly related phenomena
  - If they tell us about the same underlying variance in the data, combining them to form a single measure is reasonable
    - Parsimony
    - Reduction in Error
  - We want to combine related variables, and focus on uncorrelated or independent ones, especially those along which the observations have high variance
- We look for the phenomena underlying the observed covariance/co-dependence in a set of variables
- These phenomena are called “factors” or “principal components” or “independent components,” depending on the methods used
  - Factor analysis: based on variance/covariance/correlation
  - Independent Component Analysis: based on independence

An example:





Principal Component Analysis

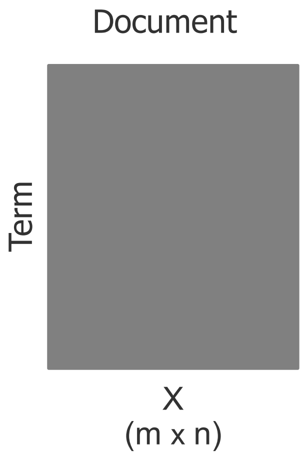
- Find a "projection direction" in which data has the maximum variance:

$$E(\Sigma_i(u^T x_i)^2) = E((u^T X)(u^T X)^T) = E(u^T X X^T u)$$

where $C = E(XX^T)$ is the covariance matrix of the data.

- So we are looking for w that maximizes $\mathbf{u}^T\mathbf{C}\mathbf{u}$, subject to $\mathbf{u}$ being unit-length

Principal Component Analysis



$$\text{Maximise} \quad u^T X X^T u$$
$$\text{s.t} \qquad u^T u = 1$$

Construct Langrangian $\qquad u^T X X^T u - \lambda u^T u$

Vector of partial derivatives set to zero

$$xx^T u - \lambda u = (xx^T - \lambda I)\, u = 0$$

As $u \neq 0$ then u must be an eigenvector of XX^T with eigenvalue λ

Eigenvalues & Eigenvectors

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

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

(right) eigenvector $\mathbf{v} \in \mathbb{R}^m \neq \mathbf{0}$

eigenvalue $\lambda \in \mathbb{R}$

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) – <u>can be complex even though S is real.</u>

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.

$$\text{if } |S - \lambda I| = 0 \text{ and } \mathrm{S} = \mathrm{S}^{\mathrm{T}} \Rightarrow \lambda \in \Re$$

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

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

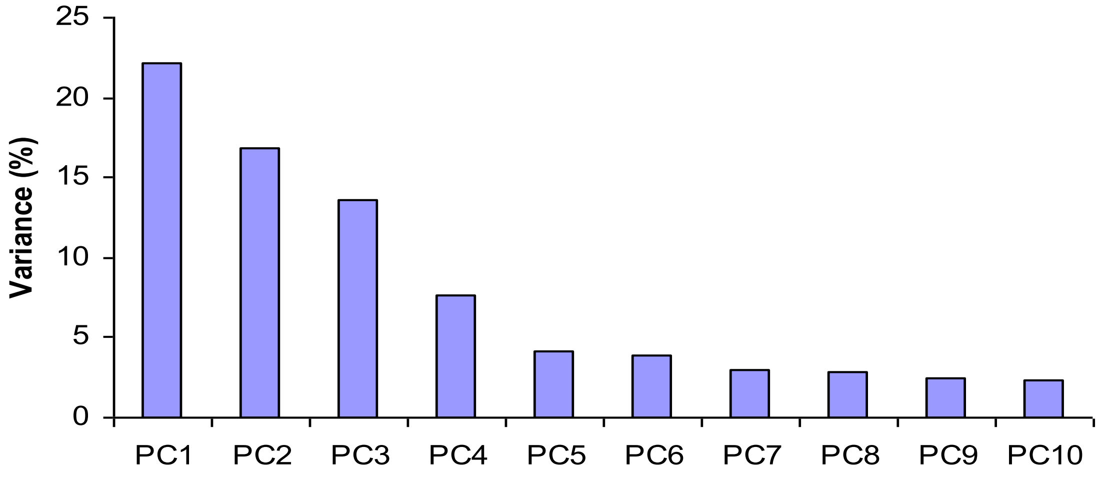
PCs, Variance and Least-Squares

- The Engen vectors are known as the “Principal Component”
- The first PC retains the greatest amount of variation in the sample
- The k^{th} PC retains the kth greatest fraction of the variation in the sample
- The k^{th} largest eigenvalue of the correlation matrix C is the variance in the sample along the k^{th} PC
- The least-squares view: PCs are a series of linear least squares fits to a sample, each orthogonal to all previous ones

How Many PCs?

- For n original dimensions, sample covariance matrix is nxn, and has up to n eigenvectors. So n PCs.
- Where does dimensionality reduction come from?

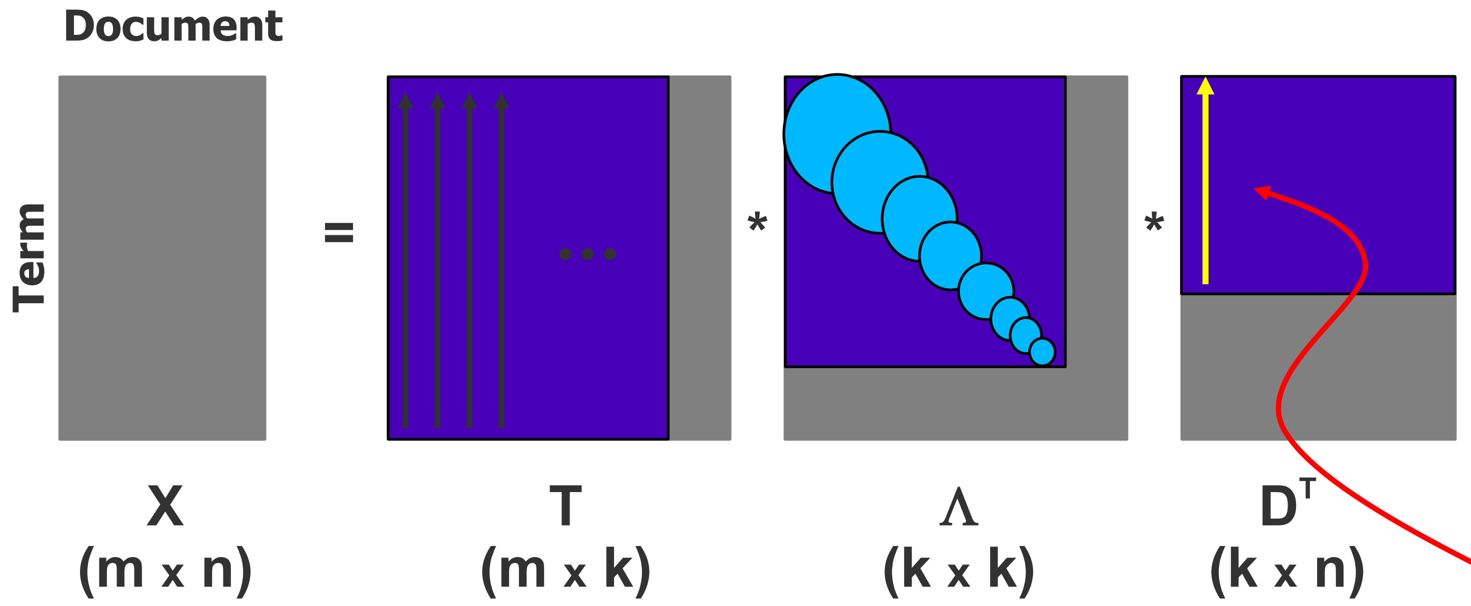
  Can *ignore* the components of lesser significance.



You do lose some information, but if the eigenvalues are small, you don't lose much

- n dimensions in original data
- calculate n eigenvectors and eigenvalues
- choose only the first p eigenvectors, based on their eigenvalues
- final data set has only p dimensions

Latent Semantic Indexing



$$\vec{w} = \sum_{k=1}^{K} d_k \lambda_k \vec{T}_k$$

This is our compressed representation of a document

Recall: Eigen/diagonal Decomposition

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

- Theorem: Exists an eigen decomposition

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

diagonal

Unique for distinct eigen-values

(cf. matrix diagonalization theorem)

- Columns of U are eigenvectors of S
- Diagonal elements of $\boldsymbol{\Lambda}$ are eigenvalues of $\mathbf{S}$

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m), \quad \lambda_i \geq \lambda_{i+1}$$

Singular Value Decomposition

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

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

$\boxed{m \times m}$ $\boxed{m \times n}$ $\boxed{V \text{ is } n \times n}$

- The columns of U are orthogonal eigenvectors of AA^T.
- The columns of V are orthogonal eigenvectors of A^TA.
- Eigenvalues $\lambda_1 \ldots \lambda_r$ of AA^T are the eigenvalues of A^TA.

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

$$\Sigma = diag(\sigma_1 \ldots \sigma_r)$$

$\boxed{\text{Singular values.}}$

SVD and PCA

- The first root is called the prinicipal eigenvalue which has an associated orthonormal ($u^T u = 1$) *eigenvector* u
- Subsequent roots are ordered such that $\lambda_1 > \lambda_2 > \ldots > \lambda_M$ with rank(D) non-zero values.
- Eigenvectors form an orthonormal basis i.e. $u_i^T u_j = \delta_{ij}$
- The eigenvalue decomposition of $XX^T = U\Sigma U^T$,

  where $U = [u_1, u_2, \ldots, u_M]$ and $\Sigma = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_M]$
- Similarly the eigenvalue decomposition of $X^T X = V\Sigma V^T$

- The SVD is closely related to the above $X = U\,\Sigma^{1/2}\,V^T$
- The left eigenvectors U, right eigenvectors V,
- singular values = square root of eigenvalues.

Example

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

U (9x7) =

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

S (7x7) =

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

V (7x8) =

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

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

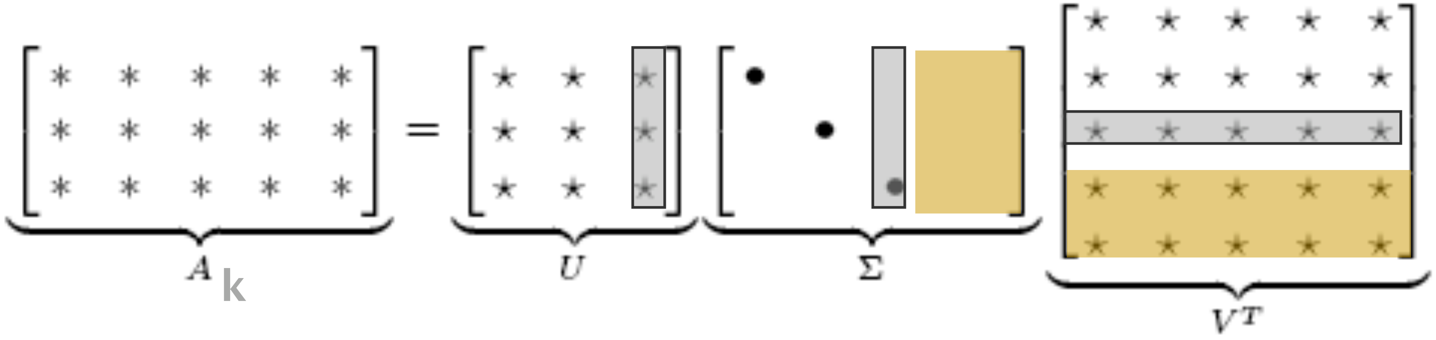
Singular values = Sqrt of Eigen values of AA^T

Low-rank Approximation

- Solution via SVD

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

set smallest r-k
singular values to zero



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

column notation: sum of rank 1 matrices

Approximation error

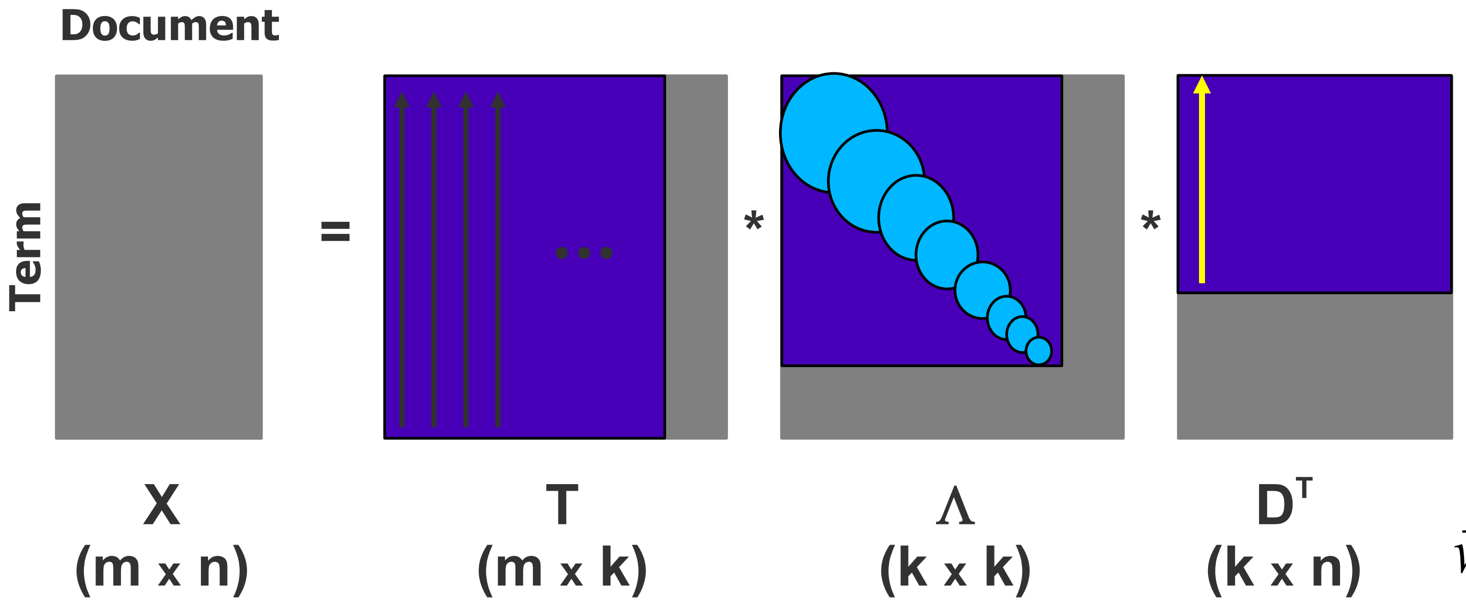
- How good (bad) is this approximation?
- It's the best possible, measured by the Frobenius norm of the error:

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

where the σ_i are ordered such that $\sigma_i \geq \sigma_{i+1}$.
Suggests why Frobenius error drops as *k* increased.

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.



$$\vec{w} = \sum_{k=1}^{K} d_k \lambda_k \vec{T}_k$$

C. Eckart, G. Young, *The approximation of a matrix by another of lower rank*. Psychometrika, 1, 211-218, 1936.

Following the Example

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

U (9x7) =

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

S (7x7) =

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

V^T (7x8) =

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

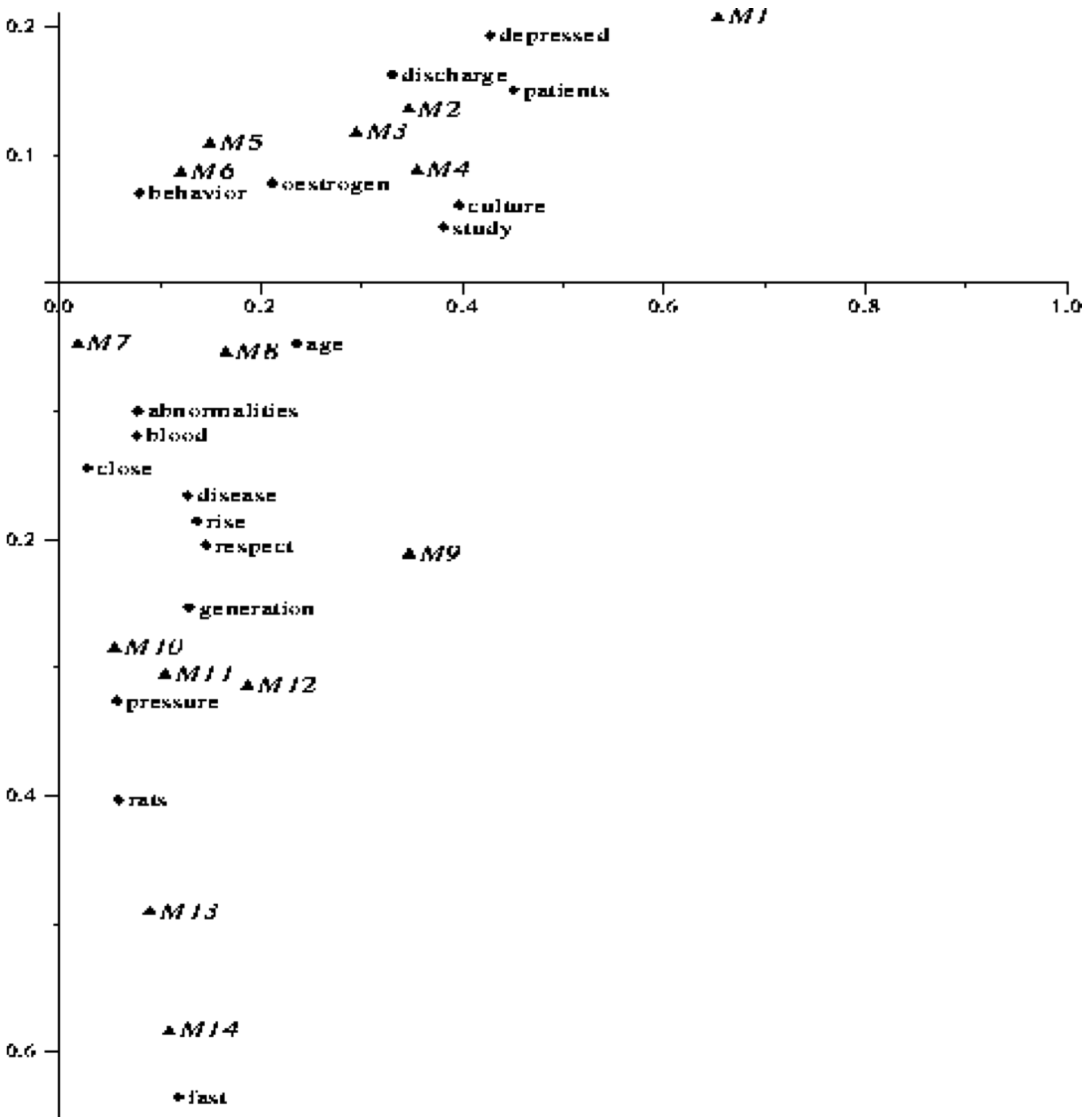
This happens to be a rank-7 matrix -so only 7 dimensions required

Singular values = Sqrt of Eigen values of AA^T

Medline data

Label	Medical Topic
M1	study of depressed patients after discharge with regard to age of onset and culture
M2	culture of pleuropneumonia like organisms found in vaginal discharge of patients
M3	study showed oestrogen production is depressed by ovarian irradiation
M4	cortisone rapidly depressed the secondary rise in oestrogen output of patients
M5	boys tend to react to death anxiety by acting out behavior while girls tended to become depressed
M6	changes in children's behavior following hospitalization studied a week after discharge
M7	surgical technique to close ventricular septal defects
M8	chromosomal abnormalities in blood cultures and bone marrow from leukaemic patients
M9	study of christmas disease with respect to generation and culture
M10	insulin not responsible for metabolic abnormalities accompanying a prolonged fast
M11	close relationship between high blood pressure and vascular disease
M12	mouse kidneys show a decline with respect to age in the ability to concentrate the urine during a water fast
M13	fast cell generation in the eye lens epithelium of rats
M14	fast rise of cerebral oxygen pressure in rats

Terms	Documents													
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
abnormalities	0	0	0	0	0	0	0	1	0	1	0	0	0	0
age	1	0	0	0	0	0	0	0	0	0	0	1	0	0
behavior	0	0	0	0	1	1	0	0	0	0	0	0	0	0
blood	0	0	0	0	0	0	0	1	0	0	1	0	0	0
close	0	0	0	0	0	0	1	0	0	0	1	0	0	0
culture	1	1	0	0	0	0	0	1	1	0	0	0	0	0
depressed	1	0	1	1	1	0	0	0	0	0	0	0	0	0
discharge	1	1	0	0	0	1	0	0	0	0	0	0	0	0
disease	0	0	0	0	0	0	0	0	1	0	1	0	0	0
fast	0	0	0	0	0	0	0	0	0	1	0	1	1	1
generation	0	0	0	0	0	0	0	0	1	0	0	0	1	0
oestrogen	0	0	1	1	0	0	0	0	0	0	0	0	0	0
patients	1	1	0	1	0	0	0	1	0	0	0	0	0	0
pressure	0	0	0	0	0	0	0	0	0	0	1	0	0	1
rats	0	0	0	0	0	0	0	0	0	0	0	0	1	1
respect	0	0	0	0	0	0	0	1	0	0	0	1	0	0
rise	0	0	0	1	0	0	0	0	0	0	0	0	0	1
study	1	0	1	0	0	0	0	0	1	0	0	0	0	0



Querying

- To query for *feedback controller*, the query vector would be

  $q = [0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]'$

  (' indicates transpose)

- Then the document-space vector corresponding to q is given by:

  $q'*U2*inv(S2) = Dq$

  - Point at the centroid of the query terms' positions in the new space.

- For the *feedback controller* query vector, the result is:

  $Dq = 0.1376 \quad 0.3678$

- To find the best document match, we compare the *Dq* vector against all the document vectors in the 2-dimensional *V2* space. The document vector that is nearest in direction to *Dq* is the best match. The cosine values for the eight document vectors and the query vector are:

- -0.3747 0.9671 0.1735 -0.9413 0.0851 0.9642 -0.7265 -0.3805

U2 (9x2) =

$$\begin{array}{rr} 0.3996 & -0.1037 \\ 0.4180 & -0.0641 \\ 0.3464 & -0.4422 \\ 0.1888 & 0.4615 \\ 0.3602 & 0.3776 \\ 0.4075 & 0.3622 \\ 0.2750 & 0.1667 \\ 0.2259 & -0.3096 \\ 0.2958 & -0.4232 \end{array}$$

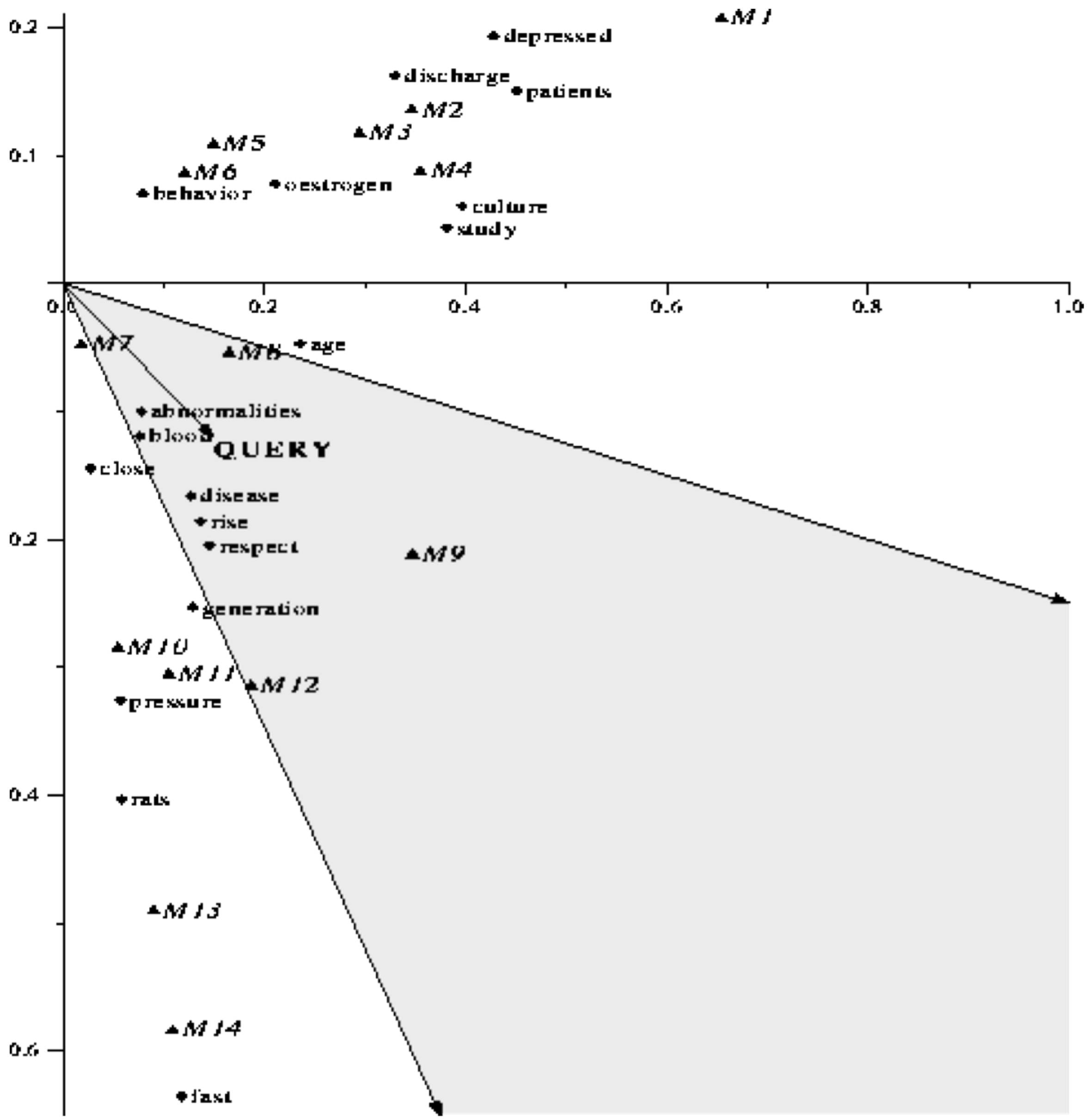
S2 (2x2) =

$$\begin{array}{rr} 3.9901 & 0 \\ 0 & 2.2813 \end{array}$$

V2 (8x2) =

$$\begin{array}{rr} 0.2917 & -0.2674 \\ 0.3399 & 0.4811 \\ 0.1889 & -0.0351 \\ -0.0000 & -0.0000 \\ 0.6838 & -0.1913 \\ 0.4134 & 0.5716 \\ 0.2176 & -0.5151 \\ 0.2791 & -0.2591 \end{array}$$

term	**ch2**	**ch3**	**ch4**	**ch5**	**ch6**	**ch7**	**ch8**	**ch9**
controllability	1	1	0	0	1	0	0	1
observability	1	0	0	0	1	1	0	1
realization	1	0	1	0	1	0	1	0
feedback	0	1	0	0	0	1	0	0
controller	0	1	0	0	1	1	0	0
observer	0	1	1	0	1	1	0	0
transfer function	0	0	0	0	1	1	0	0
polynomial	0	0	0	0	1	0	1	0
matrices	0	0	0	0	1	0	1	1
	-0.37	0.967	0.173	-0.94	0.08	0.96	-0.72	-0.38



$$\begin{pmatrix} 0.1491 & -0.1199 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}^T \begin{pmatrix} 0.1623 & -0.1372 \\ 0.2068 & -0.0488 \\ 0.0597 & 0.0614 \\ 0.1663 & -0.1313 \\ 0.0258 & -0.1246 \\ 0.4534 & 0.0386 \\ 0.3579 & 0.1710 \\ 0.2931 & 0.1426 \\ 0.0690 & -0.1576 \\ 0.0940 & -0.6535 \\ 0.0599 & -0.2378 \\ 0.1560 & 0.0661 \\ 0.4948 & 0.1091 \\ 0.0460 & -0.3393 \\ 0.0369 & -0.4196 \\ 0.1797 & -0.1456 \\ 0.1087 & -0.2126 \\ 0.3814 & 0.0941 \end{pmatrix} \begin{pmatrix} 3.5919 & 0 \\ 0 & 2.6471 \end{pmatrix}^{-1}$$

Number of Factors					
$k = 2$		$k = 4$		$k = 8$	
M 9	1.00	M 8	0.92	M 8	0.67
M12	0.88	M 9	0.89	M12	0.55
M 8	0.85	M 2	0.64	M10	0.54
M11	0.82	M10	0.48		
M10	0.79	M12	0.46		
M 7	0.74	M11	0.40		
M14	0.72				
M13	0.71				
M 4	0.67				
M 1	0.56				
M 2	0.42				

Within .40 threshold

K is the number of singular values used

What LSI can do

- LSI analysis effectively does
  - Dimensionality reduction
  - Noise reduction
  - Exploitation of redundant data
  - Correlation analysis and Query expansion (with related words)
- Some of the individual effects can be achieved with simpler techniques (e.g. thesaurus construction). LSI does them together.
- LSI handles synonymy well, not so much polysemy
- Challenge: SVD is complex to compute ($O(n^3)$)
  - Needs to be updated as new documents are found/updated

Summary:

- Principle
  - Linear projection method to reduce the number of parameters
  - Transfer a set of correlated variables into a new set of uncorrelated variables
  - Map the data into a space of lower dimensionality
  - Form of unsupervised learning
- Properties
  - It can be viewed as a rotation of the existing axes to new positions in the space defined by original variables
  - New axes are orthogonal and represent the directions with maximum variability
- Application: 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.
- Limitation: Linear projection/embedding (see supplementary)

Supplementary

Nonlinear DR, and manifold learning

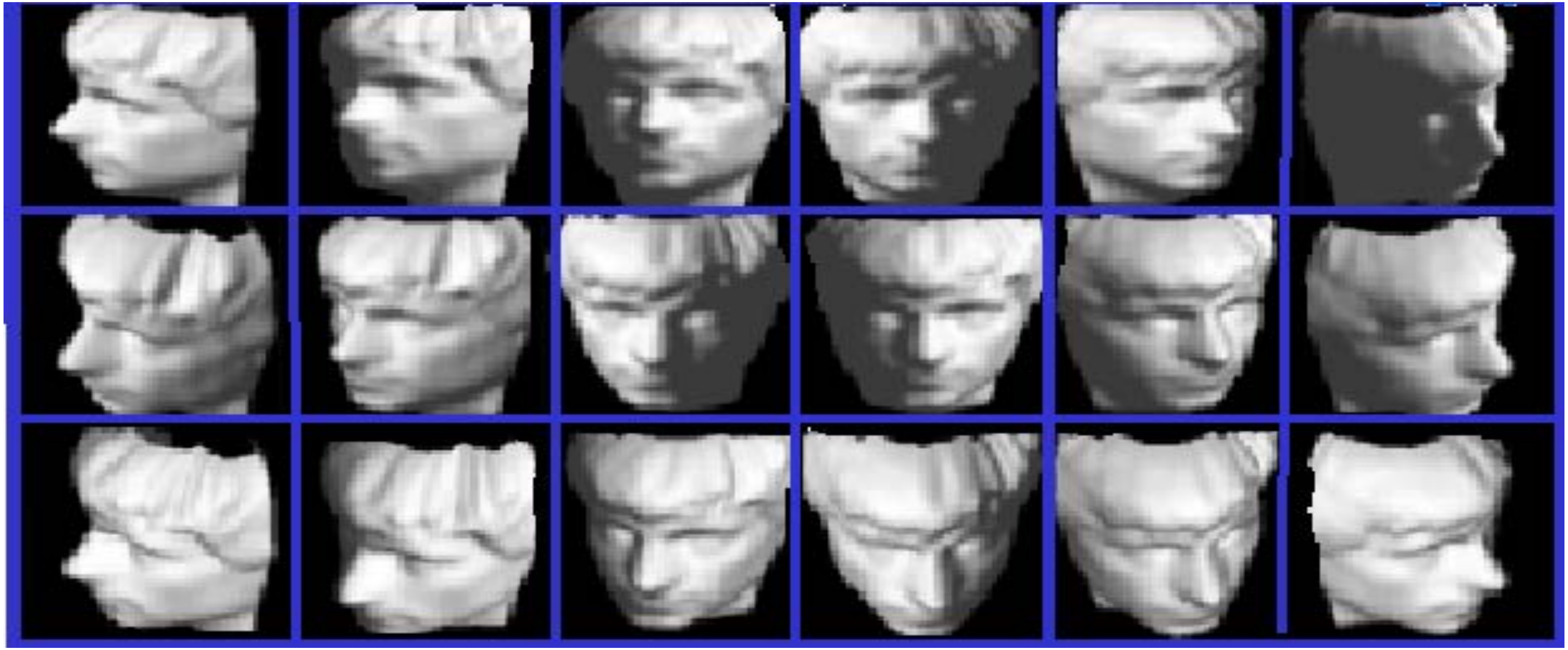
Image retrieval/labelling

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Dimensionality Bottlenecks

- Data dimension
  - Sensor response variables X:
    - 1,000,000 samples of an EM/Acoustic field on each of N sensors
    - 1024^2 pixels of a projected image on a IR camera sensor
    - N^2 expansion factor to account for all pairwise correlations
- Information dimension
  - Number of free parameters describing probability densities f(X) or f(S|X)
    - For known statistical model: info dim = model dim
    - For unknown model: info dim = dim of density approximation
- Parametric-model driven dimension reduction
  - DR by sufficiency, DR by maximum likelihood
- Data-driven dimension reduction
  - Manifold learning, structure discovery

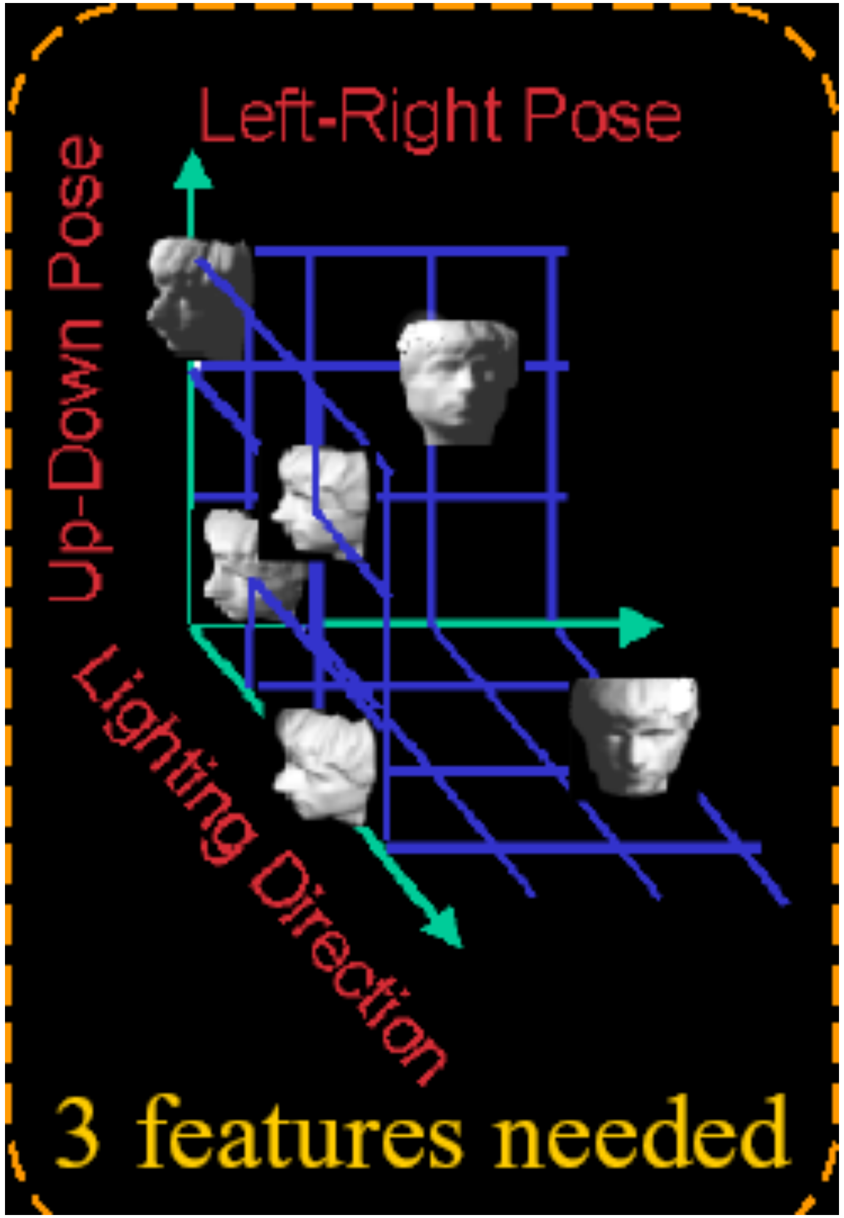
Intuition: how does your brain store these pictures?



Brain Representation

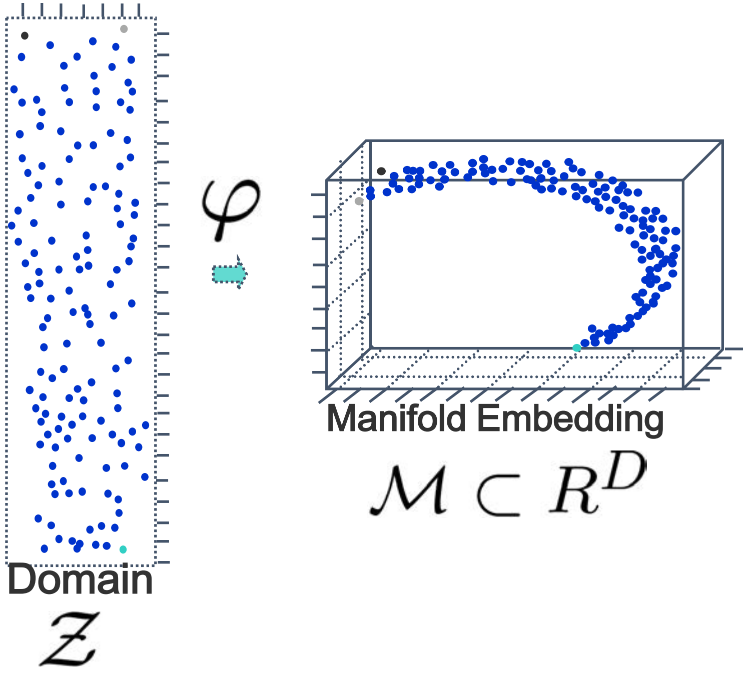
- Every pixel?
- Or perceptually meaningful structure?
  - Up-down pose
  - Left-right pose
  - Lighting direction

So, your brain successfully reduced the high-dimensional inputs to an intrinsically 3-dimensional manifold!



Two Geometries to Consider

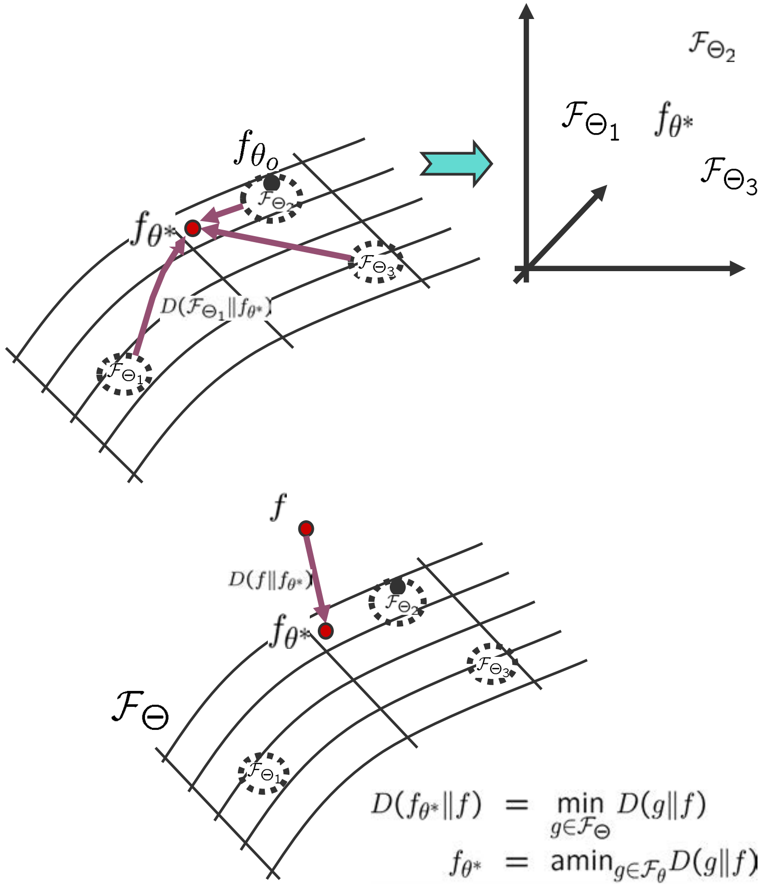
(Metric) data geometry



$\{X_i \in \mathcal{M}\}_i$ are i.i.d. samples from $f_X(x)$

$$P(X \in B) = \int_{B \cap \mathcal{M}} f_X(x)dx$$
$$= \int_{\varphi^{-1}(B \cap \mathcal{M})} f_Z(z)dz$$

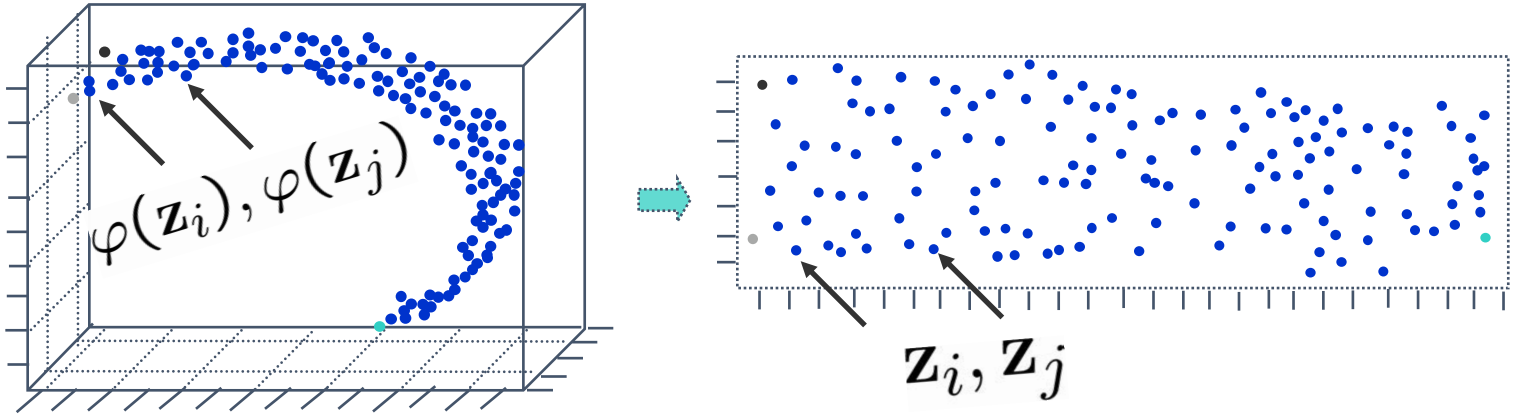
(Non-metric) information geometry



$$D(f_{\theta^*} \| f) = \min_{g \in \mathcal{F}_\Theta} D(g \| f)$$
$$f_{\theta^*} = \mathrm{amin}_{g \in \mathcal{F}_\theta} D(g \| f)$$

Data-driven DR

- Data-driven projection to lower dimensional subsapce
- Extract low-dim structure from high-dim data
- Data may lie on curved (but locally linear) subspace



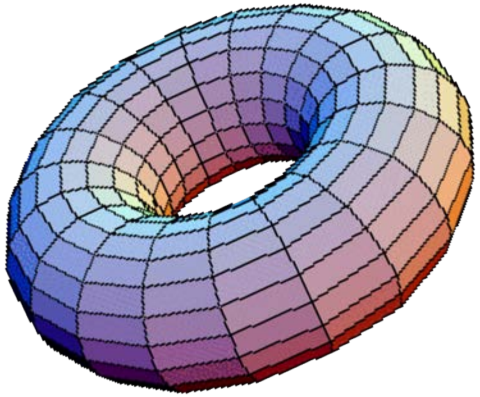
[1] Josh .B. Tenenbaum, Vin de Silva, and John C. Langford “A Global Geometric Framework for Nonlinear Dimensionality Reduction” *Science*, 22 Dec 2000.

[2] Jose Costa, Neal Patwari and Alfred O. Hero, “Distributed Weighted Multidimensional Scaling for Node Localization in Sensor Networks”, *IEEE/ACM Trans. Sensor Networks*, to appear 2005.

[3] Misha Belkin and Partha Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation*, 2003.

What is a Manifold?

- A manifold is a topological space which is locally Euclidean.
- Represents a very useful and challenging unsupervised learning problem.
- In general, any object which is nearly "flat" on small scales is a manifold.



Going beyond

- What is the essence of the C matrix?

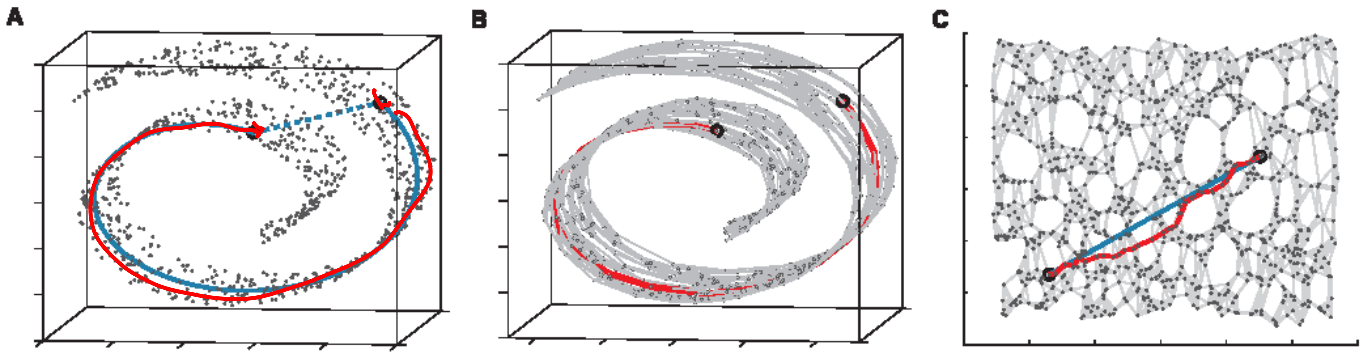
$$C = E[XX^T] = \frac{1}{n}\mathbf{X}\mathbf{X}^T$$

- The elements in C captures some kind of affinity between a pair of data points in the semantic space

- We can replace it with any reasonable affinity measure
  - E.g., $D = \left(\|x_i - x_j\|^2\right)_{ij}$: distance matrix MDS
  - E.g., the geodistance ISOMAP

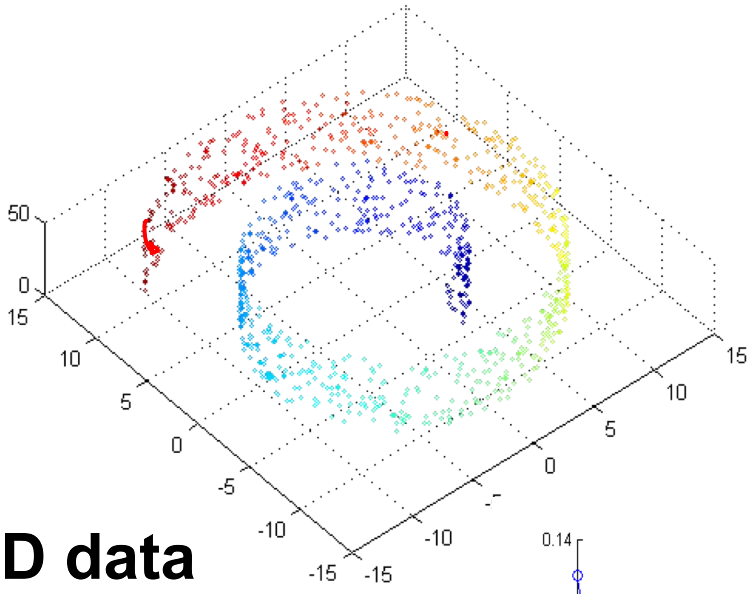
Nonlinear DR – Isomap

[Josh. Tenenbaum, Vin de Silva, John langford 2000]

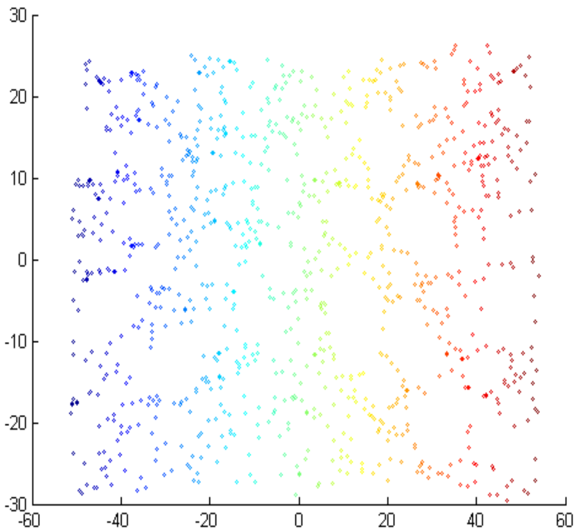


- Constructing neighbourhood graph G
- For each pair of points in G, Computing shortest path distances ---- geodesic distances.
  - Use Dijkstra's or Floyd's algorithm
- Apply kernel PCA for C given by the centred matrix of squared geodesic distances.
- Project test points onto principal components as in kernel PCA.

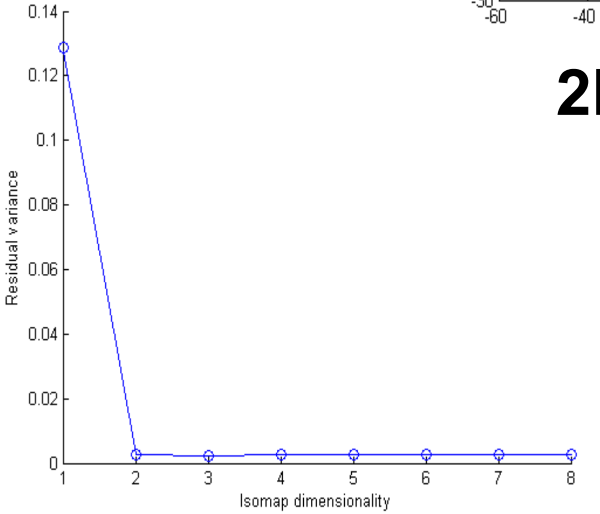
“Swiss Roll” dataset



3D data



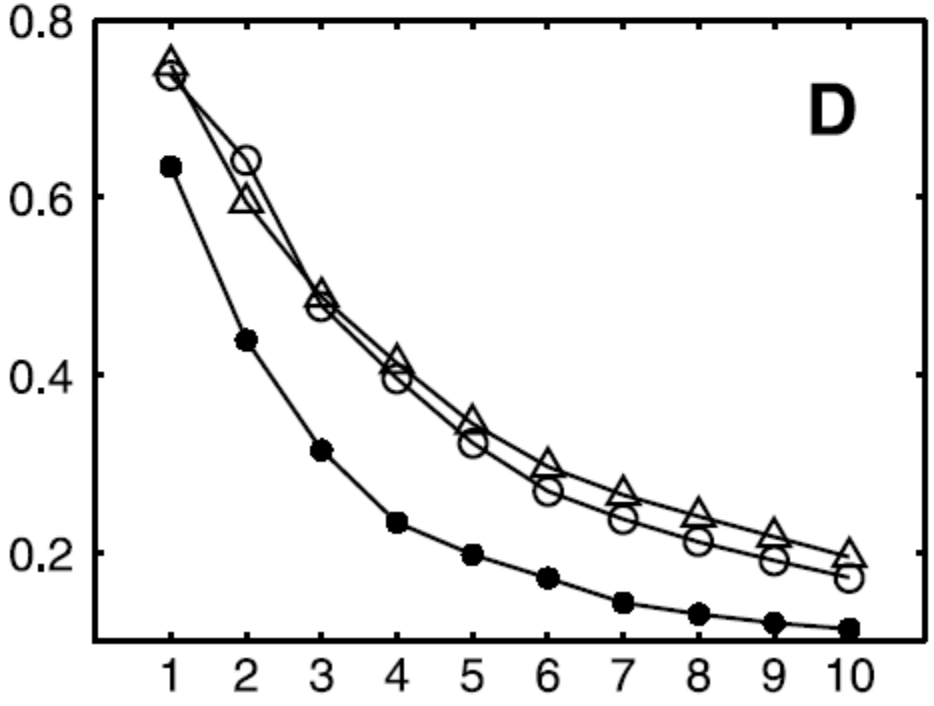
2D coord chart

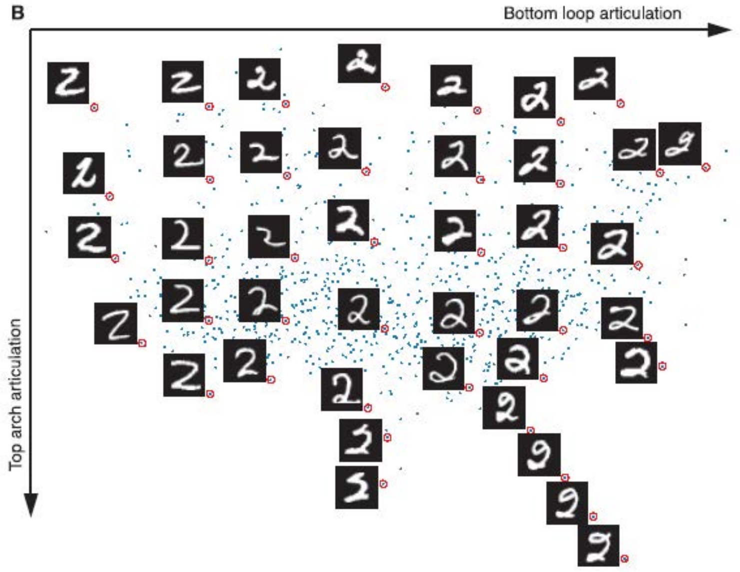


Error vs. dimensionality of coordinate chart

PCA, MD vs ISOMAP

- The residual variance of PCA (open triangles), MDS (open circles), and Isomap





ISOMAP algorithm Pros/Cons

Advantages:

- Nonlinear
- Globally optimal
- Guarantee asymptotically to recover the true dimensionality

Drawback:

- May not be stable, dependent on topology of data
- As N increases, pair wise distances provide better approximations to geodesics, but cost more computation

Local Linear Embedding (a.k.a LLE)

- LLE is based on simple geometric intuitions.

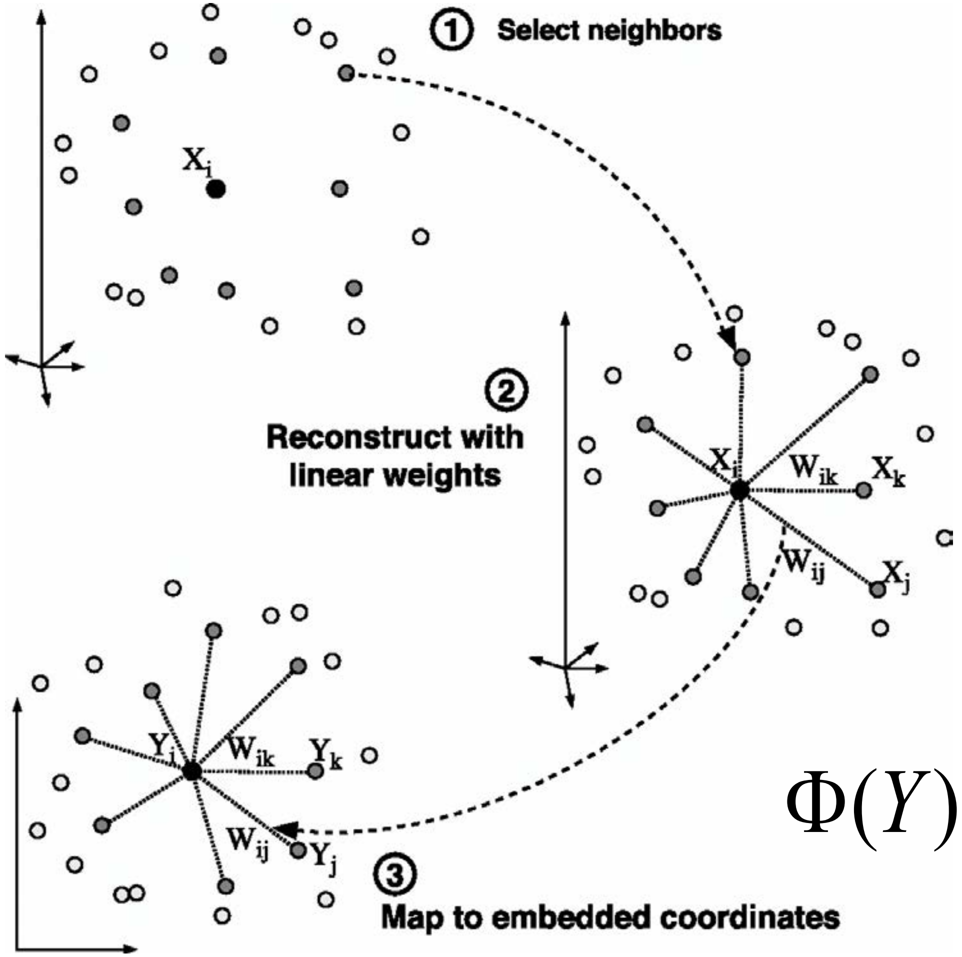
- Suppose the data consist of N real-valued vectors X_i, each of dimensionality D.

- Each data point and its neighbors expected to lie on or close to a locally linear patch of the manifold.

Steps in LLE algorithm

- Assign neighbors to each data point $\vec{X}_i$
- Compute the weights W_{ij} that best linearly reconstruct the data point from its neighbors, solving the constrained least-squares problem.
- Compute the low-dimensional embedding vectors $\vec{Y}_i$ best reconstructed by W_{ij}.

Fit locally, Think Globally



From Nonlinear Dimensionality Reduction by Locally Linear Embedding

Sam T. Roweis and Lawrence K. Saul

$$\Phi(Y) = \sum_i |\vec{Y} - \sum_j W_{ij} \vec{Y}_j|^2$$

Super-Resolution Through Neighbor Embedding

[Yeung et al CVPR 2004]

Training Xs^i

img1.jpg

Training Ys^i

Testing Xt

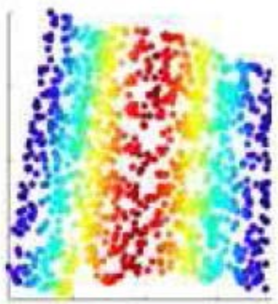
?

Testing Yt

Intuition

- Patches of the image lie on a manifold

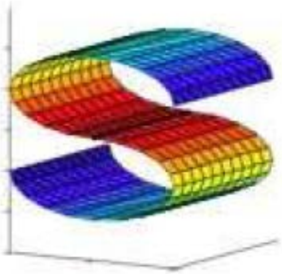
Training



Low dimensional Manifold

img1.jpg

Training Ys[i]



High dimensional Manifold

Algorithm

1. Get feature vectors for each low resolution training patch.
2. For each test patch feature vector find K nearest neighboring feature vectors of training patches.
3. Find optimum weights to express each test patch vector as a weighted sum of its K nearest neighbor vectors.
4. Use these weights for reconstruction of that test patch in high resolution.

Results

Training Xs^i

Training Ys^i

Testing Xt

Testing Yt

Summary:

- Principle
  - Linear and nonlinear projection method to reduce the number of parameters
  - Transfer a set of correlated variables into a new set of uncorrelated variables
  - Map the data into a space of lower dimensionality
  - Form of unsupervised learning

- Applications
  - PCA and Latent semantic indexing for text mining
  - Isomap and Nonparametric Models of Image Deformation
  - LLE and Isomap Analysis of Spectra and Colour Images
  - Image Spaces and Video Trajectories: Using Isomap to Explore Video Sequences
  - Mining the structural knowledge of high-dimensional medical data using isomap

Isomap Webpage: http://isomap.stanford.edu/