Machine Learning

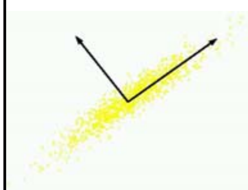
10-701/15-781, Spring 2008

Principal Components Analysis

Modified from www.cs.princeton.edu/picasso/mats/Lecture1_jps.ppt

Eric Xing

Lecture 24, April 16, 2008



Reading: Chap 12.1, CB book

Factor or Component Analysis: Why?

- 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

Factor or Component Analysis, Why? (cond.)

- 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 physical meaning is obscure

The goal:

- Discover a new set of factors/dimensions/axes based on which to represent, describe or evaluate the data
  - For more effective reasoning, insights, or better visualization
  - Reduce noise in the data
  - Typically a smaller set of factors: dimension reduction
  - Better representation of data without losing much information
  - Can build more effective data analyses on the reduced-dimensional space: classification, clustering, pattern recognition

- Factors are combinations of observed variables
  - May be more effective bases for insights, even if physical meaning is obscure
  - Observed data are described in terms of these factors rather than in terms of original variables/dimensions

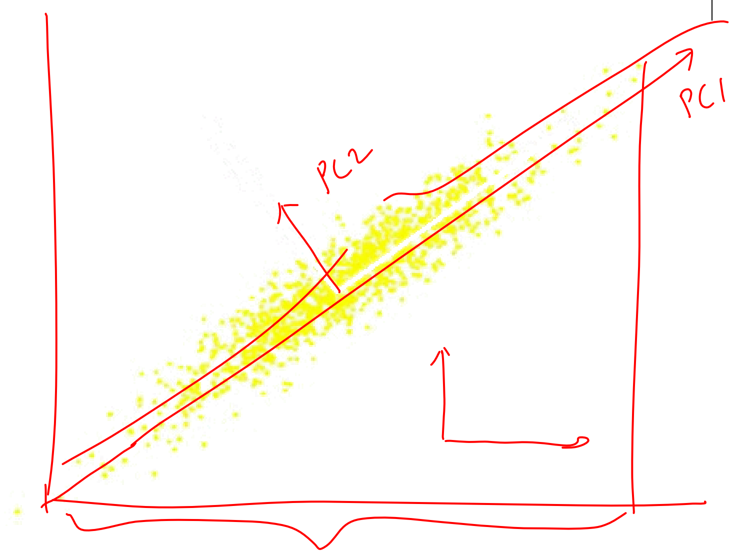
Basic Concept

- Areas of variance in data are where items can be best discriminated and key underlying phenomena observed
  - Areas of greatest "signal" in the data
- 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
- So we want to combine related variables, and focus on uncorrelated or independent ones, especially those along which the observations have high variance
- We want a smaller set of variables that explain most of the variance in the original data, in more compact and insightful form

Basic Concept

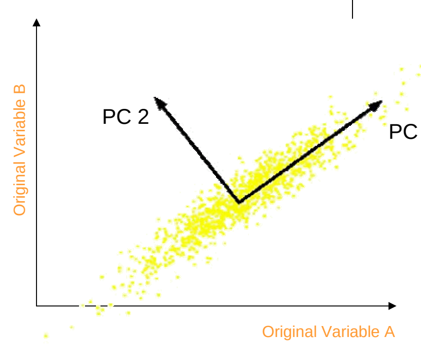
- What if the dependences and correlations are not so strong or direct?
- And suppose you have 3 variables, or 4, or 5, or 10000?
- Look for the phenomena underlying the observed covariance/co-dependence in a set of variables
  - Once again, phenomena that are uncorrelated or independent, and especially those along which the data show high variance
- 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:



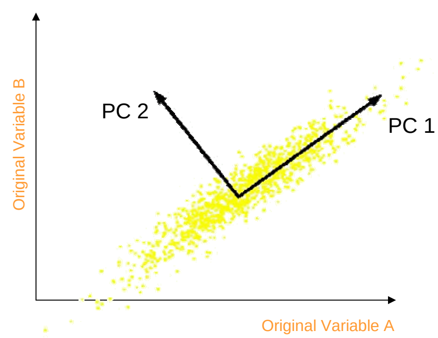
Principal Component Analysis

- Most common form of factor analysis
- The new variables/dimensions
  - Are linear combinations of the original ones
  - Are uncorrelated with one another
    - Orthogonal in original dimension space
  - Capture as much of the original variance in the data as possible
  - Are called Principal Components



- Orthogonal directions of greatest variance in data
- Projections along PC1 discriminate the data most along any one axis

Principal Component Analysis



- First principal component is the direction of greatest variability (covariance) in the data
- Second is the next orthogonal (uncorrelated) direction of greatest variability
  - So first remove all the variability along the first component, and then find the next direction of greatest variability
- And so on …

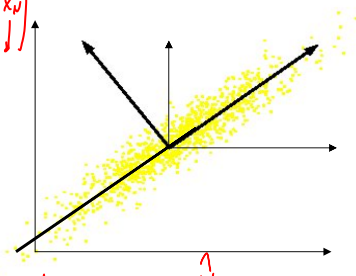
Computing the Components

- Data points are vectors in a multidimensional space
- Projection of vector **x** onto an axis (dimension) **u** is $\mathbf{u}^T\mathbf{x}$
- Direction of greatest variability is that in which the average square of the projection is greatest
  - I.e. **u** such that $E((\mathbf{u}^T\mathbf{x})^2)$ over **all x** is maximized
  - Matrix representation:

$$\begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ x_1 & x_2 & \cdots & x_N \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}$$

$$\text{proj} = u^T \left(X \right) = (p_1\ p_2 \cdots\ )$$

  - (we subtract the mean along each dimension, and center the original axis system at the centroid of all data points, for simplicity)
  - This direction of **u** is the direction of the first Principal Component



$$x_i = \hat{x}_i - \mu. \qquad \mu = \frac{\sum \hat{x}_i}{N}$$

Computing the Components

- $E(\Sigma_i(\mathbf{u}^T\mathbf{x}_i)^2) = E((\mathbf{u}^T\mathbf{X})(\mathbf{u}^T\mathbf{X})^T) = E(\mathbf{u}^T\mathbf{X}\mathbf{X}^T\mathbf{u})$
- The **covariance matrix** $\mathbf{C} = \mathbf{X}\mathbf{X}^T$ contains the correlations (similarities) of the original axes based on how the data values project onto them
- So we are looking for w that maximizes $\mathbf{u}^T\mathbf{C}\mathbf{u}$, subject to $\mathbf{u}$ being unit-length

$\max_u u^T C u.$ s.t. $u^T u = 1$.

- It is maximized when w is the principal eigenvector of the matrix **C**, in which case
  - $\mathbf{u}^T\mathbf{C}\mathbf{u} = \mathbf{u}^T\lambda\mathbf{u} = \lambda$ if $\mathbf{u}$ is unit-length, where λ is the principal eigenvalue of the correlation matrix C
  - The eigenvalue denotes the amount of variability captured along that dimension

$Cu = \lambda u \qquad u^* = u_1 \qquad \lambda_1 > \lambda_2 \cdots \lambda_M.$

Why the Eigenvectors?

Maximise $\mathbf{u}^T\mathbf{X}\mathbf{X}^T\mathbf{u}$

s.t $\mathbf{u}^T\mathbf{u} = 1$

Construct Langrangian $\mathbf{u}^T\mathbf{X}\mathbf{X}^T\mathbf{u} - \lambda\mathbf{u}^T\mathbf{u}$

Vector of partial derivatives set to zero

$$\mathbf{x}\mathbf{x}^T\mathbf{u} - \lambda\mathbf{u} = (\mathbf{x}\mathbf{x}^T - \lambda\mathbf{I})\,\mathbf{u} = 0$$

As $\mathbf{u} \neq \mathbf{0}$ then $\mathbf{u}$ must be an eigenvector of $\mathbf{X}\mathbf{X}^T$ with eigenvalue λ

Eigenvalues & Eigenvectors

- **Eigenvectors** (for a square $m \times m$ matrix 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 **S** is real.

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**.

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

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

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

Eigen/diagonal Decomposition

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

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

$U = [u_1 \; u_2 \; \cdots \; u_m]$

Unique for distinct eigen-values

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

diagonal

(cf. matrix diagonalization theorem)

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

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

Computing the Components

- Similarly for the next axis, etc.
- So, the new axes are the eigenvectors of the matrix of correlations of the original variables, which captures the similarities of the original variables based on how data samples project to them

- Geometrically: centering followed by rotation
  - Linear transformation

PCs, Variance and Least-Squares

$(\ ) \Leftarrow u u_i^T = (\ )(\ )$

- 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

$C = XX^T = \sum_{i=1}^{m} \lambda u_i u_i^T$

- The least-squares view: PCs are a series of linear least squares fits to a sample, each orthogonal to all previous ones

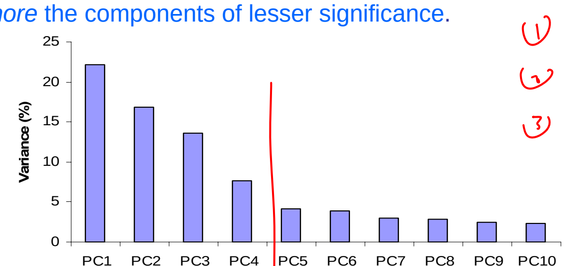
$\approx \sum_{i=1}^{k} \lambda_i u_i u_i^T$

$\lambda_1 > \lambda_2 \cdots \lambda_m$

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.



(1) Eigen Gap

(2) Amount of loss.

(3) $k = 2$ or 3

$x_i = \begin{pmatrix} x_{i1} \\ \vdots \\ x_{im} \end{pmatrix} \Rightarrow y = (\ )_3^1$

$y_1 = u_1^T x$

$y_2 = u_2^T x$

$y_3 \cdots$

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

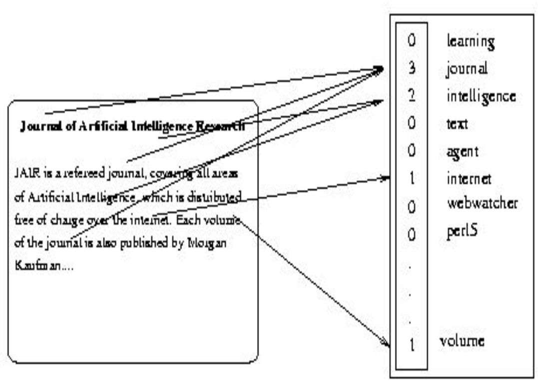
- n dimensions in original data
- calculate n eigenvectors and eigenvalues
- choose only the first p eigenvectors, based on their eigenvalues
- final data set has only p dimensions

Application: Latent Semantic Analysis

- Motivation
  - Lexical matching at term level inaccurate (claimed)
  - Polysemy – words with number of 'meanings' – term matching returns irrelevant documents – impacts precision
  - Synonomy – number of words with same 'meaning' – term matching misses relevant documents – impacts recall
- LSA assumes that there exists a LATENT structure in word usage – obscured by variability in word choice
- Analogous to signal + additive noise model in signal processing

The Vector Space Model

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



The Corpora Matrix

X =

	Doc 1	Doc 2	Doc 3	...
Word 1	3	0	0	...
Word 2	0	8	1	...
Word 3	0	1	3	...
Word 4	2	0	0	...
Word 5	12	0	0	...
...	0	0	0	...

Feature Vector Representation

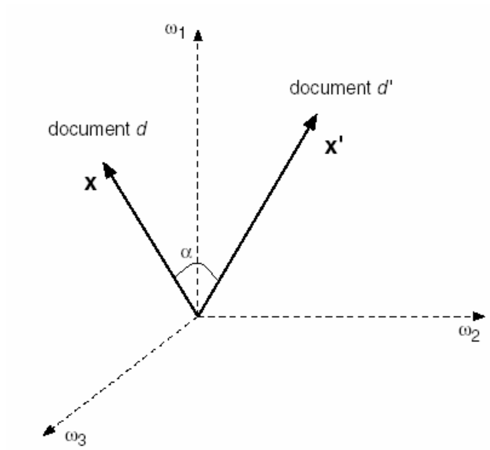


Figure 4.2 Cosine measure of document similarity.

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.

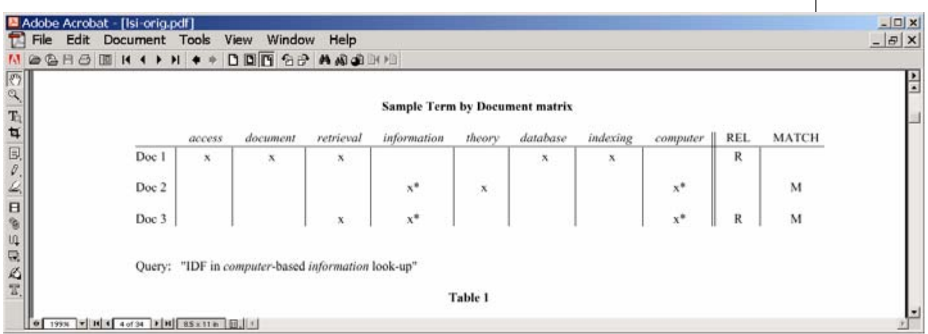
$$\mathrm{sim}_{\mathrm{true}}(d,q) > \cos(\angle(\vec{d},\vec{q}))$$

  - **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.

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

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

Example of Problems



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

Latent Semantic Indexing (LSI)

(Deerwester et al., 1990)

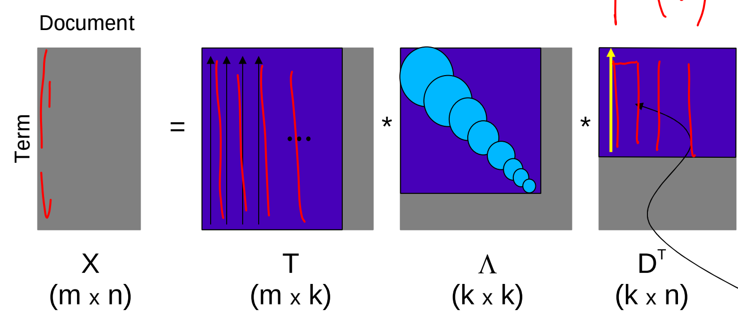
- Uses statistically derived conceptual indices instead of individual words for retrieval
- Assumes that there is some underlying or *latent* structure in word usage that is obscured by variability in word choice
- Key idea: instead of representing documents and queries as vectors in a t-dim space of 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 to reduce document representations,
  - The axes are the Principal Components from SVD
- So what is SVD?

$$\begin{array}{c|c} & \text{Tech.} \quad \text{Agri} \quad \cdots \\ \hline \begin{array}{c}\text{Intel}\\ \text{Micro}\\ \text{Apple}\\ \text{banana}\\ \text{pear}\\ \vdots\end{array} & \begin{bmatrix} 2.5 & 0.1 & \\ 3.9 & 0.1 & \\ 2.8 & 3.2 & \\ 0.1 & 3 & 2 \\ 0.2 & 2.8 & \\ \vdots & & \end{bmatrix} \end{array}$$

Example

- Suppose we have keywords
  - Car, automobile, driver, elephant
- We want queries on car to also get docs about drivers and automobiles, but not about elephants
  - What if we could discover that the cars, automobiles and drivers axes are strongly correlated, but elephants is not
  - How? Via correlations observed through documents
  - If docs A & B don't share any words with each other, but both share lots of words with doc C, then A & B will be considered similar
  - E.g A has cars and drivers, B has automobiles and drivers
- When you scrunch down dimensions, small differences (noise) gets glossed over, and you get desired behavior

Latent Semantic Indexing



This is our compressed representation of a document

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

Recall: Eigen/diagonal decomposition

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

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

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

diagonal

Unique for distinct eigen-values

(cf. matrix diagonalization theorem)

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

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

Singular Value Decomposition

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

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

$m \times m$ | $m \times n$ | V is $n \times n$

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

The columns of $\boldsymbol{V}$ are orthogonal eigenvectors of $\boldsymbol{A^TA}$.

m×m

n×n

Eigenvalues $\lambda_1 \ldots \lambda_r$ of $\boldsymbol{AA^T}$ are the eigenvalues of $\boldsymbol{A^TA}$.

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

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

Singular values.

SVD and PCA

- The first root is called the prinicipal eigenvalue which has an associated orthonormal ($\mathbf{u}^T\mathbf{u} = 1$) *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. $\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$
- The eigenvalue decomposition of $\mathbf{XX}^T = \mathbf{U\Sigma U}^T$
- where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]$ and $\mathbf{\Sigma} = \text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_M]$
- Similarly the eigenvalue decomposition of $\mathbf{X}^T\mathbf{X} = \mathbf{V\Sigma V}^T$
- The SVD is closely related to the above $\mathbf{X} = \mathbf{U}\, \mathbf{\Sigma}^{1/2}\, \mathbf{V}^T$
- 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) = T

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

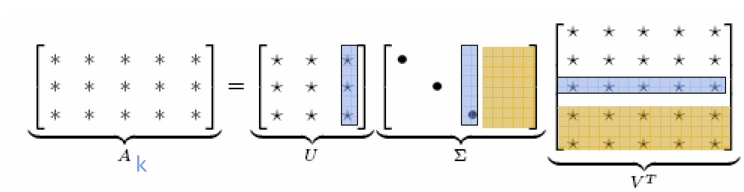
This happens to be a rank-7 matrix
-so only 7 dimensions required

Singular values = Sqrt of Eigen values of AA^T

Low-rank Approximation

- Solution via SVD

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



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

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:

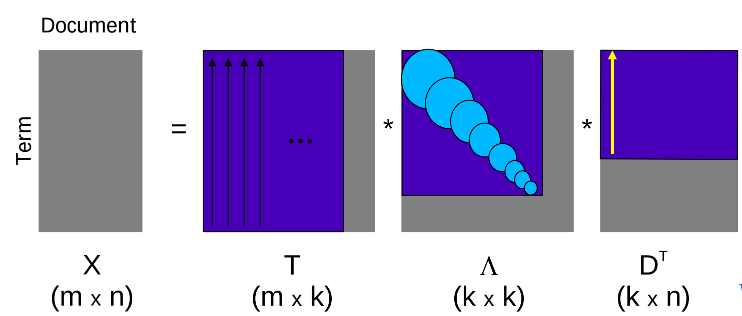
$$\min_{X:rank(X)=k} \|A-X\|_F = \|A-A_k\|_F = \sigma_{k+1}$$

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) =

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

S (7x7) =

$$\begin{bmatrix} 3.9901 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.2813 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.6705 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.3522 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.1818 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.6623 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.6487 \end{bmatrix}$$

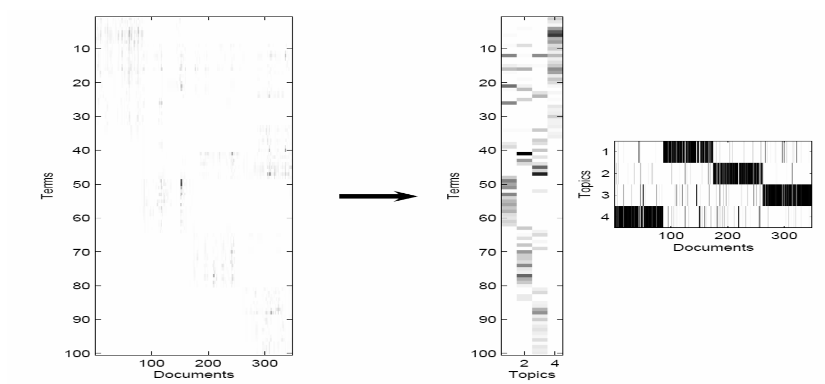
V (7x8) =

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

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

Singular values = Sqrt of Eigen values of AA^T

PCs can be viewed as Topics



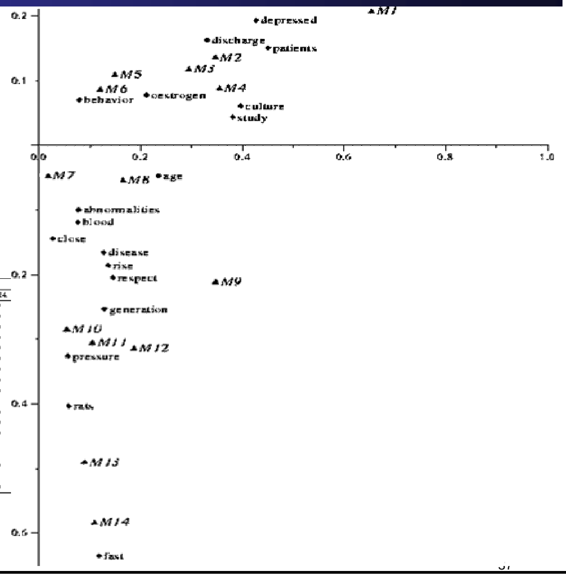
In the sense of having to find quantities that are not observable directly

Similarly, transcription factors in biology, as unobservable causal bridges between experimental conditions and gene expression

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

U2 (9x2) =
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

S2 (2x2) =
3.9901 0
0 2.2813

V2 (8x2) =
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

To query for *feedback controller*, the query vector would be
q = [0 0 0 1 1 0 0 0]' (' indicates transpose),

Let *q* be the query vector. Then the document-space vector corresponding to *q* is given by:

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

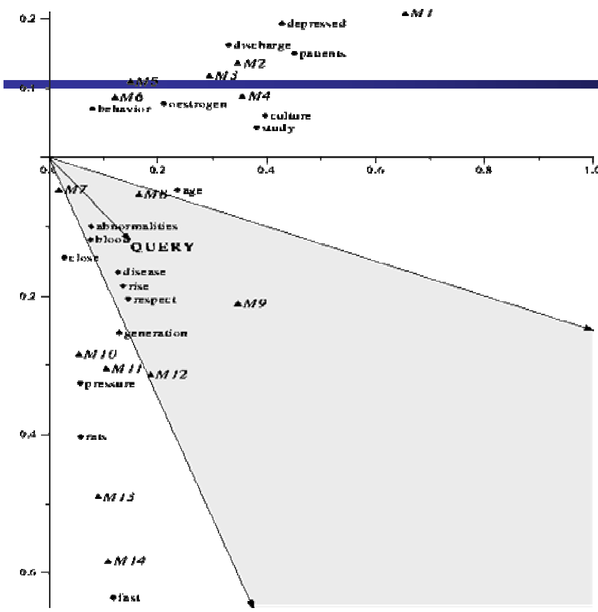
Point at the centroid of the query terms' poisitions in the new space.
For the *feedback controller* query vector, the result is:

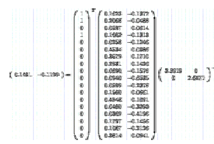
Dq = 0.1376 0.3678

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

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





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.