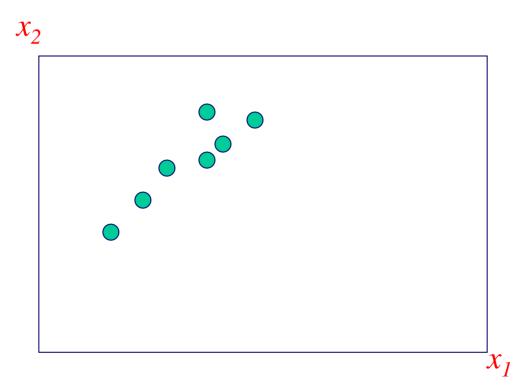
Component Analysis and Discriminants

Reducing dimensionality when:

- Classes are disregarded
 Principal Component Analysis (PCA)
- 2. Classes are considered Discriminant Analysis



Component Analysis vs Discriminants

Two classical approaches for finding effective linear transformations

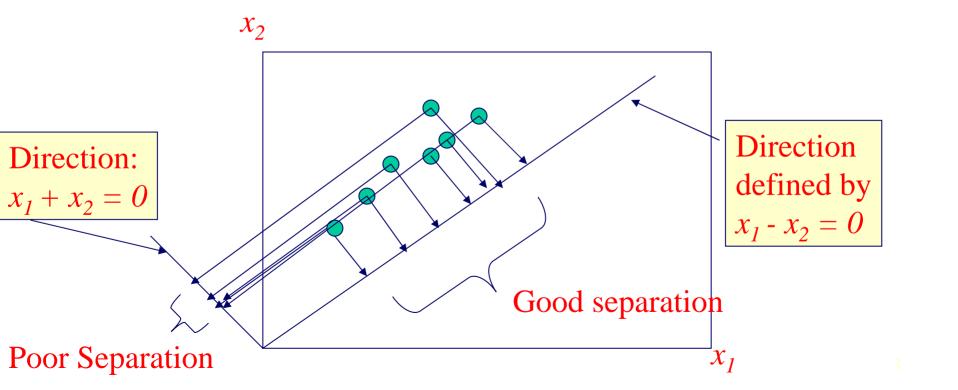
- PCA (Principal Component Analysis) "Projection that best represents data in least- square sense"
- MDA (Multiple Discriminant Analysis) "Projection that best separates the data in a least-squares sense"

PCA: Linear Projections of Data

- Excessive dimensionality $\mathbf{x} = [x_1, x_2, \dots x_d]$ causes
 - Computational difficulty
 - Visualization issues
- Solution:
 - Combine features to reduce dimensionality
 - Linear combinations, e.g., $2x_1+3x_2+x_3$
 - are simple to compute and tractable
 - Project high dimensional data onto a lower dimensional space

Projection to lower dimensional space

Allow computer to search for interesting directions



Linear Projection

• Equation of line through origin: $x_1 + 2x_2 + 4x_3 = 0$

$$\begin{bmatrix} 1 & 2 & 4 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_1 + 2x_2 + 4x_3 = 0$$

Can be written as $\mathbf{w}^t \mathbf{x} = 0$ where $\mathbf{w} = [1\ 2\ 4]$

Projection of a point x along line with projection

weights a is given by:

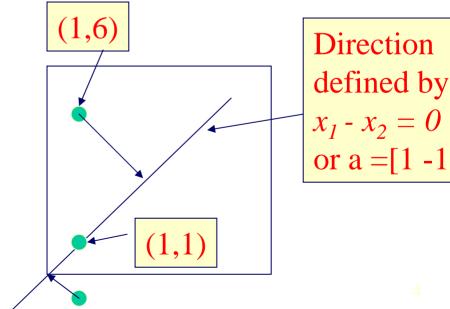
$$\mathbf{a}^T \mathbf{x} = \sum_{j=1}^d \mathbf{a}_j x_j$$

• Example:

Projection of (1,6) along [1 -1]: 1 - 6 = -5

Projection of (1,1) along [1 -1]: 1 - 1 = 0

Projection of (1,-1) along [1 -1]: 1 + 1 = 2

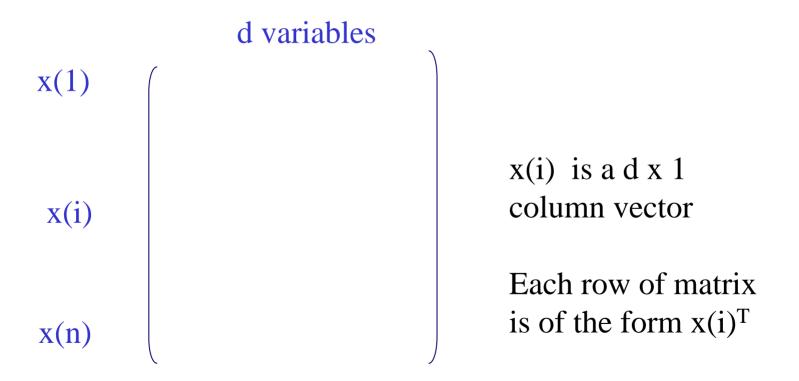


Principal Components Analysis

 We will look at Data Matrix, Scatter Matrix and Covariance Matrix to arrive at best projection of data

Data Matrix

Let X be a *n* x *d* data matrix of *n* samples



Assume X is mean-centered, so that the value of each variable is subtracted for that variable

Scatter Matrix

$$S = \sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m}) (\mathbf{x}_k - \mathbf{m})^t$$
a $d \times d$ matrix since $[d \times 1][1 \times d]$ is $[d \times d]$

- Scatter Matrix is (n-1) times Covariance Matrix
- Relationship Between Data Matrix and Scatter Matrix:

 $S = X^{t}X$ is the $d \times d$ scatter matrix of the data since X has zero mean

Projection

Let \mathbf{a} be a $p \times 1$ column vector of projection weights that result in the largest variance when the data matrix \mathbf{X} are projected along \mathbf{a}

The projection of any data vector x is the linear combination

$$\mathbf{a}^T \mathbf{x} = \sum_{j=1}^d \mathbf{a}_j x_j$$

Projected values of all data vectors in data matrix X onto a can be expressed as

Xa

which is an $n \times 1$ column vector.

Variance along Projection

Variance along a is

$$\sigma_{\mathbf{a}}^{2} = (X\mathbf{a})^{T} (X\mathbf{a})$$
$$= \mathbf{a}^{T} X^{T} X \mathbf{a}$$
$$= \mathbf{a}^{T} S \mathbf{a}$$

where $S = X^{t}X$ is the $d \times d$ covariance matrix of the data since X has zero mean

Thus variance is a function of both **a** and **S**

Maximizing variance along **a** is not well-defined since we can increase it without limit by increasing the size of the components of **a**.

Optimization Problem

Impose a normalization constraint on the **a** vectors such that $\mathbf{a}^{T}\mathbf{a} = 1$

Optimization problem is to maximize

Variance Criterion

$$u = \mathbf{a}^T S \mathbf{a} - \lambda (\mathbf{a}^T \mathbf{a} - 1)$$

Where λ is a Lagrange multiplier.

Normalization Criterion

Solution: Differentiating wrt a yields

$$\frac{\partial u}{\partial a} = 2Sa - 2\lambda a = 0$$

which reduces to

$$(S - \lambda I)a = 0$$

Characteristic Equation of S!

Characteristic Equation

Given a $d \times d$ matrix M, a very important class of linear equations is of the form $Mx = \lambda x$

dxd dx1 dx1

which can be rewritten as $(M - \lambda I)x = 0$

If M is real and symmetric there are d possible solution vectors (vectors x that satisfy the charecteristic equation) called Eigen Vectors, e_1 , ..., e_d and associated Eigen values

$$\lambda_1, ..., \lambda_d$$

First Principal Component

The matrix M is the Covariance matrix S,

Characteristic Equation

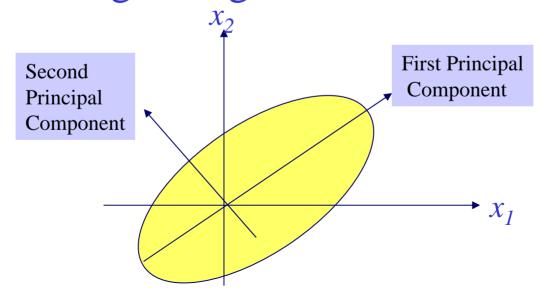
$$(S - \lambda I)a = 0$$

Roots are Eigen Values Corresponding Eigen Vectors are principal components

First principal component is the Eigen Vector associated with the largest Eigen value of S.

Other Principal Components

- Second Principal component is in direction orthogonal to first
- Has second largest Eigen value, etc



Alternative Derivation of PCA

- DHS Text gives another way to derive the fact that eigen vectors of scatter matrix are principal components
- By setting up the squared error criterion function of the data and the vector e that minimizes it satisfies the characteristic equation
- Can be generalized to from one-dimensional projection to a dimension d' which are the eigen vectors of S forming the principal components

Projection into k Eigen Vectors

Variance of data projected into first k Eigen vectors is

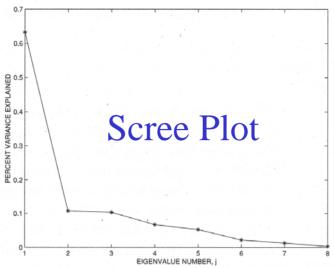
$$\sum_{j=1}^{k} \lambda_{j}$$

Squared error in approximating true data matrix X using only first k Eigen vectors is

$$\frac{\sum_{j=k+1}^{p} \lambda_{j}}{\sum_{l=1}^{p} \lambda_{l}}$$

Usually 5 to 10 principal components capture 90% of variance in the data

Example of PCA

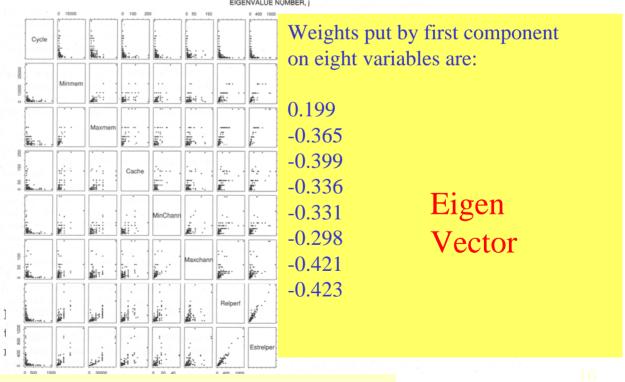


Amount of variance explained by each consecutive Eigen value

CPU data (8 variables)

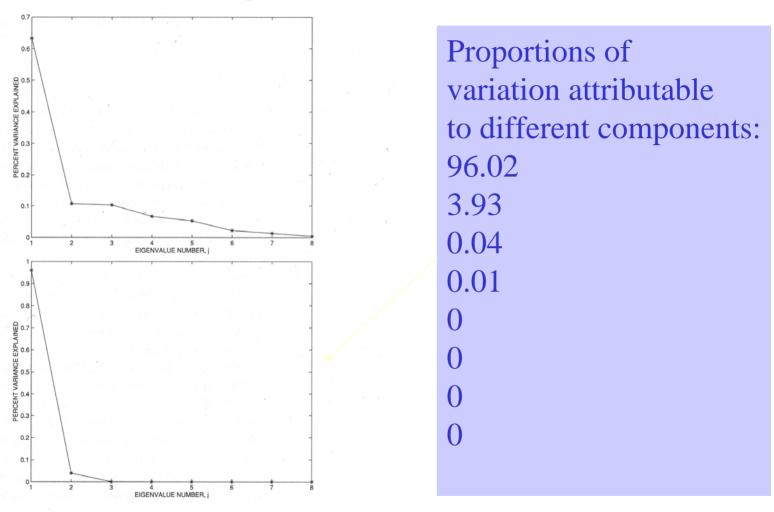
Eigen values of Correlation Matrix

CPU data Eigen values: 63.26 10.70 10.30 6.68 5.23 2.18 1.31 0.34



Scatterplot Matrix for CPU data

PCA using covariance matrix

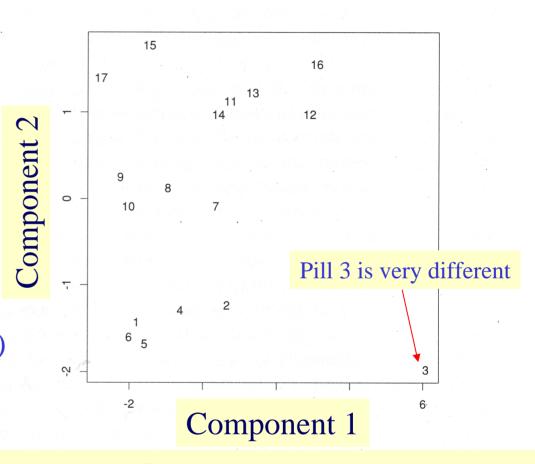


Scree Plots: (a) Eigen values from correlation matrix

(b) Eigen values from covariance matrix

Graphical Use of PCA

First two principal components of six dimensional data (17 pills: times at which specified proportion of pill has dissolved: 10%,30%,50%,70%,75%,90%)



Projection onto first two principal components

Text Retrieval Application of PCA

- Retrieving Documents based on Vector Space Model (Key Words)
- Vector Space representation of documents

	database	SQL	index	regression	likelihood	linear
D1	24	21	9	0	0	3
D2	32	10	5	0	3	0
D3	12	16	5	0	0	0
D4	6	7	2	0	0	0
D5	43	31	20	0	3	0
D6	2	0	0	18	7	16
D7	0	0	1	32	12	0
D8	3	0	0	22	4	2
D9	1	0	0	34	27	25
D10	6	0	0	17	4	23

- Query is represented as a vector, Distance is Cosine of angle between query and document
- Data Matrix is represented in terms of Principal Components using Singular Value Decomposition

Latent Semantic Indexing (LSI)

- Disadvantage of exclusive use of representing a document as a T-dimensional vector of term weights
 - Users may pose queries using terms different from terms used to index a document
 - E.g., term *data mining* is semantically similar to *knowledge discovery*

LSI method

- Approximate the *T*-dimensional term space by *k* principal components directions in this space
 - Using the *N* x *T* document term matrix to estimate directions
 - Results in a N x k matrix
 - Terms *database*, *SQL*, *indexing* etc are combined into a single principal component

Singular Value Decomposition

Document-Term Matrix, M

	database	SQL	index	regression	likelihood	linear
D1	24	21	9	0	0	3
D2	32	10	5	0	3	0
D3	12	16	5	0	0	0
D4	6	7	2	0	0	0
D5	43	31	20	0	3	0
D6	2	0	0	18	7	16
D7	0	0	1	32	12	0
D8	3	0	0	22	4	2
D9	1	0	0	34	27	25
D10	6	0	0	17	4	23

Find a decomposition $M = USV^T$

U is a 10 x 6 matrix of weights (each row for a particular document)
S is a 6 x 6 diagonal matrix of Eigen values
Columns of 6 x 6 matrix V^T represent
principal components (or orthogonal bases)

S matrix has diagonal elements 77.4, 69.5, 22.9, 13.5, 12.1, 4.8

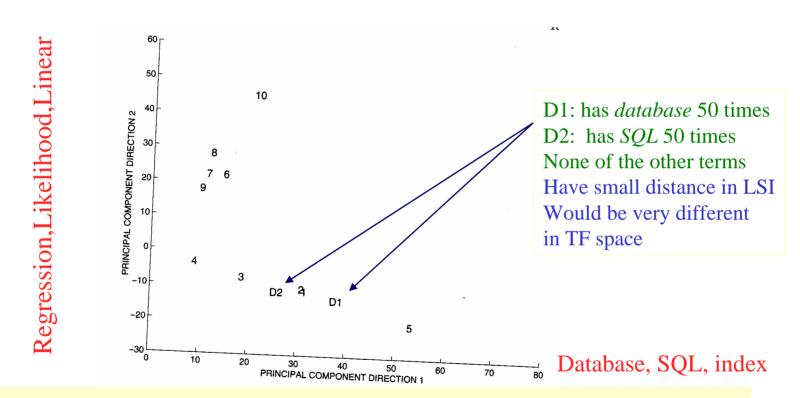
U Matrix (using 2 PCs)

Document	PC1	PC2	
d1	30.8998	-11.4912	
d2	30.3131	-10.7801	
d3	18.0007	-7.7138	
d4	8.3765	-3.5611	
d5	52.7057	-20.6051	
d6	14.2118	21.8263	
d7	10.8052	21.914	
d8	11.508	28.0101	
d9	9.5259	17.7666	
d10	19.9219	45.0751	

V matrix

	database	SQL	index	regression	likelihood	linear
v1	0.74	0.49	0.27	0.28	0.18	0.19
v2	-0.28	-0.24	-0.12	0.74	0.37	0.31

LSI Method: First Two Principal Components of Document Term Matrix



Projected locations of the 10 documents in two-dimensional plane spanned by the first two principal components of document term matrix M

LSI Practical Issues

- Query is represented as a vector in PCA space and angle calculated
 - E.g., Query SQL is converted into pseudo vector
- In practice, computing PCA vectors directly is computationally infeasible.
 - Special purpose sparse SVD techniques for highdimensions are used
- Can also model Document-Term matrix probabilistically as a mixture of component distributions of terms conditioned on topics