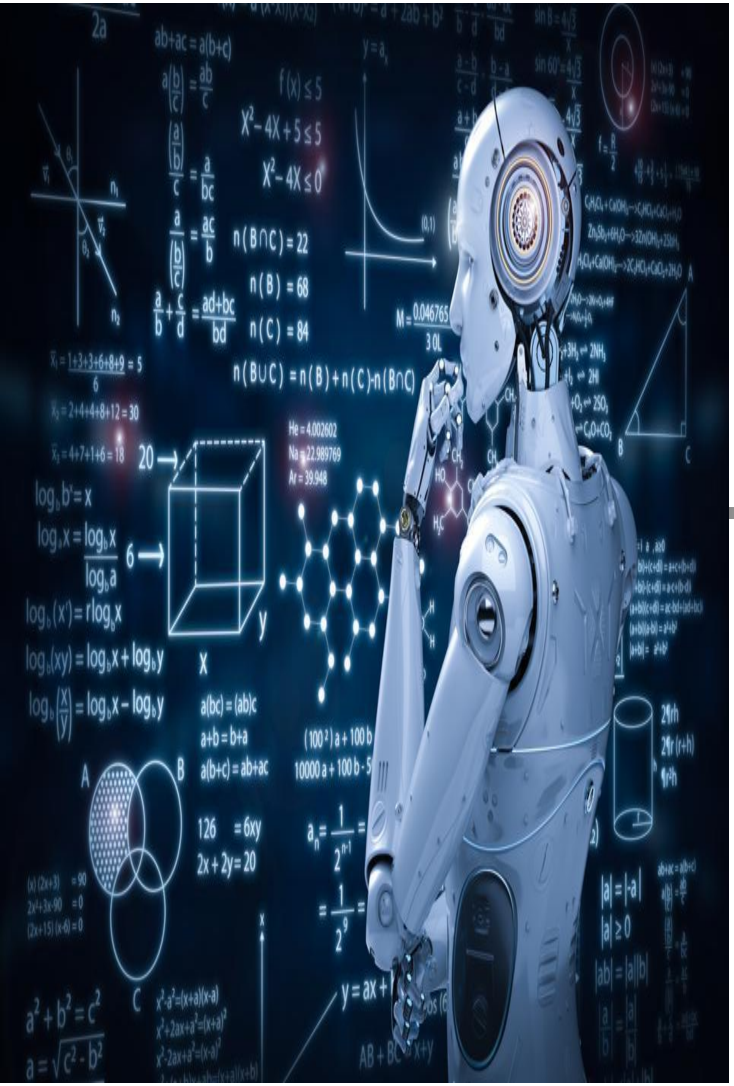


Practical Machine Learning

Day 12: SEP23 DBDA

Kiran Waghmare

Agenda

- Association
  - Apriori
  - Market Basket Analysis

`Basket data'

A very common type of data; often also called *transaction data*.

Next slide shows example *transaction database*, where each record represents a transaction between (usually) a customer and a shop.

Each record in a supermarket's transaction DB, for example, corresponds to a basket of specific items.

ID	apples,	beer,	cheese,	dates,	eggs,	fish,	glue,	honey,	ice-cream
1	1	1		1			1	1	
2			1	1	1				
3		1	1			1			
4		1				1			1
5					1		1		
6						1			1
7	1			1				1	
8						1			1
9			1		1				
10		1					1		
11					1		1		
12	1								
13			1			1			
14			1			1			
15								1	1
16				1					
17	1					1			
18	1	1	1	1				1	
19	1	1		1			1	1	
20					1				

Data

Transactions	Items
1	A B C D
2	A C D
3	A B C
4	C D E
5	A B C E

Matrix representation

Transactions	A	B	C	D	E
1	1	1	1	1	0
2	1	0	1	1	0
3	1	1	1	0	0
4	0	0	1	1	1
5	1	1	1	0	1

Execution of Apriori algorithm, $\varepsilon = 1$

Iteration 1

Candidates of size 1	*Support*
A	4
B	3
C	5
D	3
~~E~~	~~1~~

Iteration 2

Candidates of size 2	*Support*
A B	3
A C	4
A D	2
B C	3
~~B D~~	~~1~~
C D	3

Iteration 3

Candidates of size 3	*Support*
A B C	3
~~A B D~~	~~1~~
A C D	2

$$Rule:\ X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

$$Support = \frac{P(A \cap B)}{n}$$

$$Confidence = \frac{P(A \cap B)}{P(A)}$$

$$Lift = \frac{P(A \cap B)}{P(A) \bullet P(B)}$$

Support ($A \Rightarrow B$) = $P(A \cap B)$

Confidence ($A \Rightarrow B$) =$P(B|A)$

Lift ($A \Rightarrow B$) = $P(B|A)/P(B)$

Discovering Rules

A common and useful application of data mining

A `rule' is something like this:

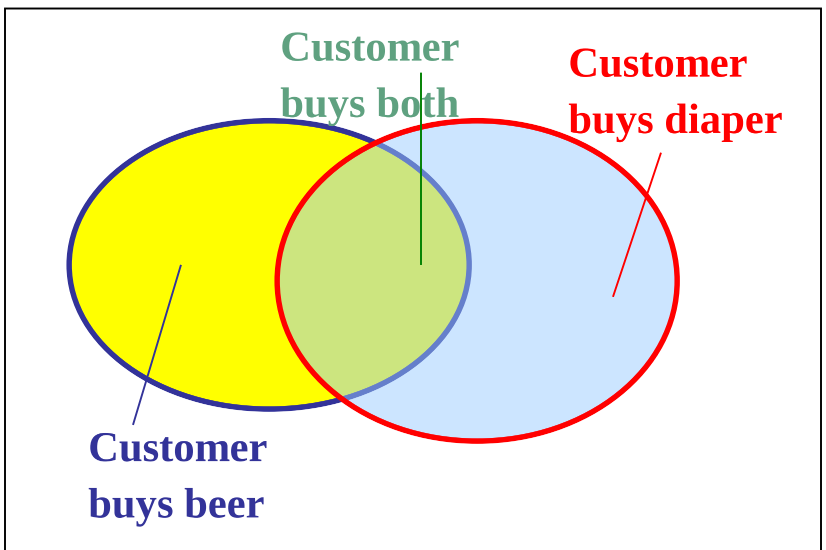
If a basket contains apples and cheese, then it also contains beer

Any such rule has two associated measures:

1. *confidence* – when the `if' part is true, how often is the `then' bit true? This is the same as *accuracy*.
2. *coverage* or *support* – how much of the database contains the `if' part?

Basic Concepts: Frequent Patterns

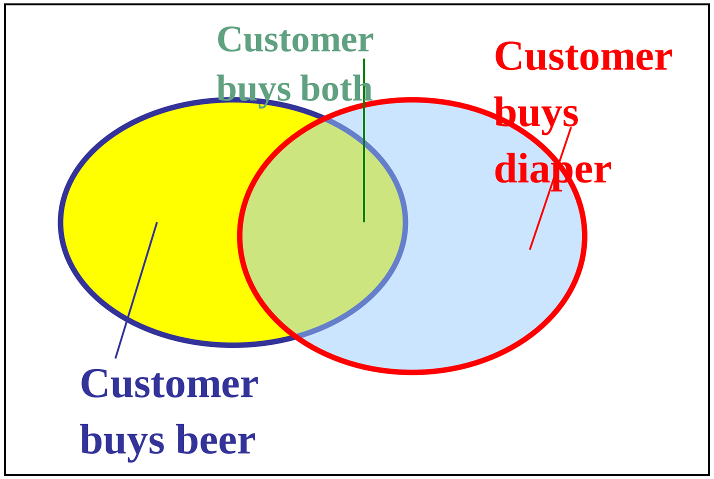
Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- itemset: A set of one or more items
- k-itemset $X = \{x_1, \ldots, x_k\}$
- *(absolute) support*, or, *support count* of X: Frequency or occurrence of an itemset X
- *(relative) support*, *s*, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X)
- An itemset X is *frequent* if X's support is no less than a *minsup* threshold

Basic Concepts: Association Rules

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, *s*, probability that a transaction contains $X \cup Y$
  - confidence, *c*, conditional probability that a transaction having X also contains *Y*

Let minsup = 50%, minconf = 50%

Freq. Pat.: Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

- Association rules: (many more!)
  - *Beer* → *Diaper* (60%, 100%)
  - *Diaper* → *Beer* (60%, 75%)

Item set	Sup-count
Hot Dogs	4
Buns	2
Ketchup	2
Coke	3
Chips	4

Item set	Sup-count
Hot Dogs	4
Buns	2
Ketchup	2
Coke	3
Chips	4

Item set	Sup-count
Hot Dogs, Buns	2
Hot Dogs, Ketchup	1
Hot Dogs, Coke	2
Hot Dogs, Chips	2
Buns, Ketchup	1
Buns, Coke	0
Buns, Chips	0
Ketchup, Coke	0
Ketchup, Chips	1
Coke, Chips	3

Item set	Sup-count
Hot Dogs, Buns	2
Hot Dogs, Coke	2
Hot Dogs, Chips	2
Coke, Chips	3

Item set	Sup-count
Hot Dogs, Buns, Coke	0
Hot Dogs, Buns, Chips	0
Hot Dogs, Coke, Chips	2

Item set	Sup-count
Hot Dogs, Coke, Chips	2

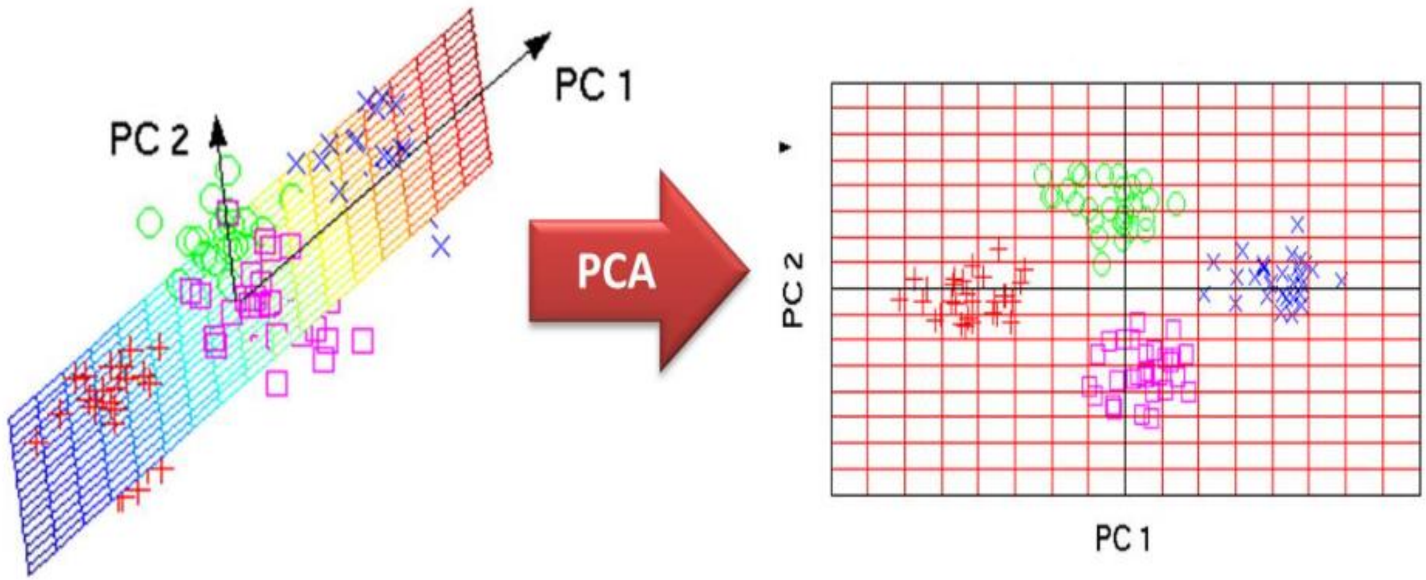
What Is Frequent Pattern Analysis?

- Frequent pattern: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of frequent itemsets and association rule mining
- Applications
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis.

Why Is Freq. Pattern Mining Important?

- Freq. pattern: An intrinsic and important property of datasets
- Foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: discriminative, frequent pattern analysis
  - Cluster analysis: frequent pattern-based clustering
  - Data warehousing: iceberg cube and cube-gradient
  - Semantic data compression: fascicles
  - Broad applications

Dimensionality Reduction & Principal Component Analysis



PCA

Dimensionality reduction

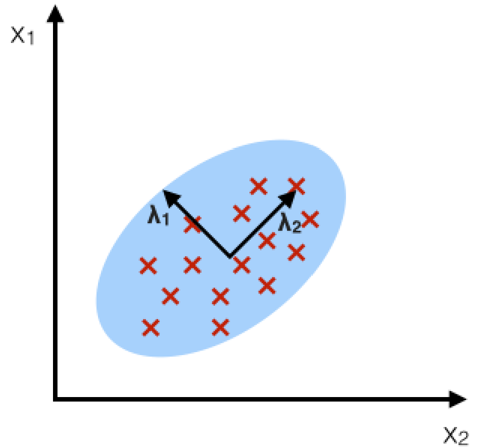
- Many modern data domains involve huge numbers of features / dimensions
  - Documents: thousands of words, millions of bigrams
  - Images: thousands to millions of pixels
  - Genomics: thousands of genes, millions of DNA polymorphisms

Why reduce dimensions?

- High dimensionality has many costs
  - Redundant and irrelevant features degrade performance of some ML algorithms
  - Difficulty in interpretation and visualization
  - Computation may become infeasible
    - what if your algorithm scales as $O(n^3)$?
  - Curse of dimensionality

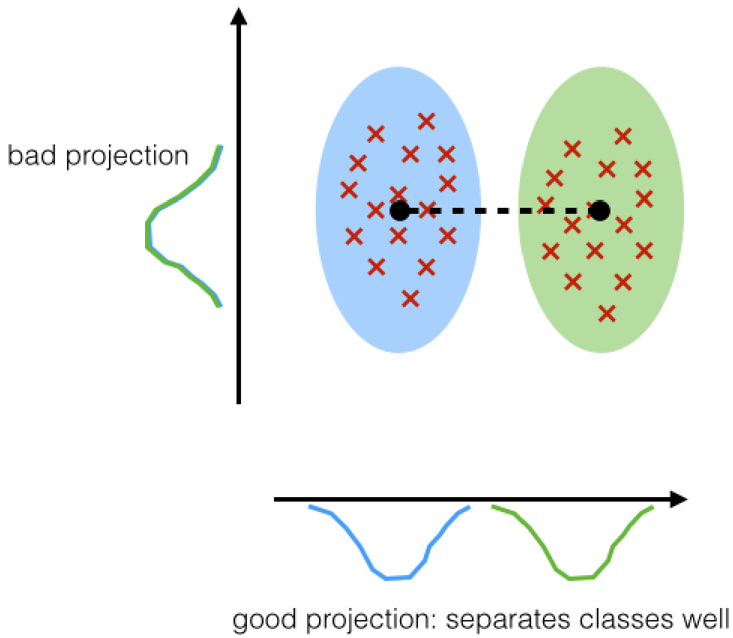
PCA:

component axes that maximize the variance



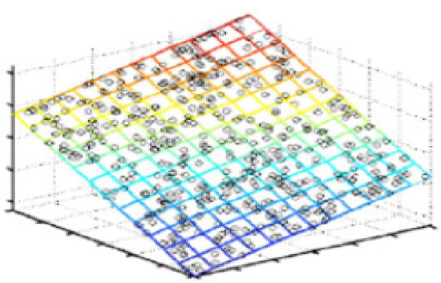
LDA:

maximizing the component axes for class-separation



Unsupervised dimensionality reduction

- Consider data without class labels
- Try to find a more compact representation of the data



$3d \Rightarrow 2d$

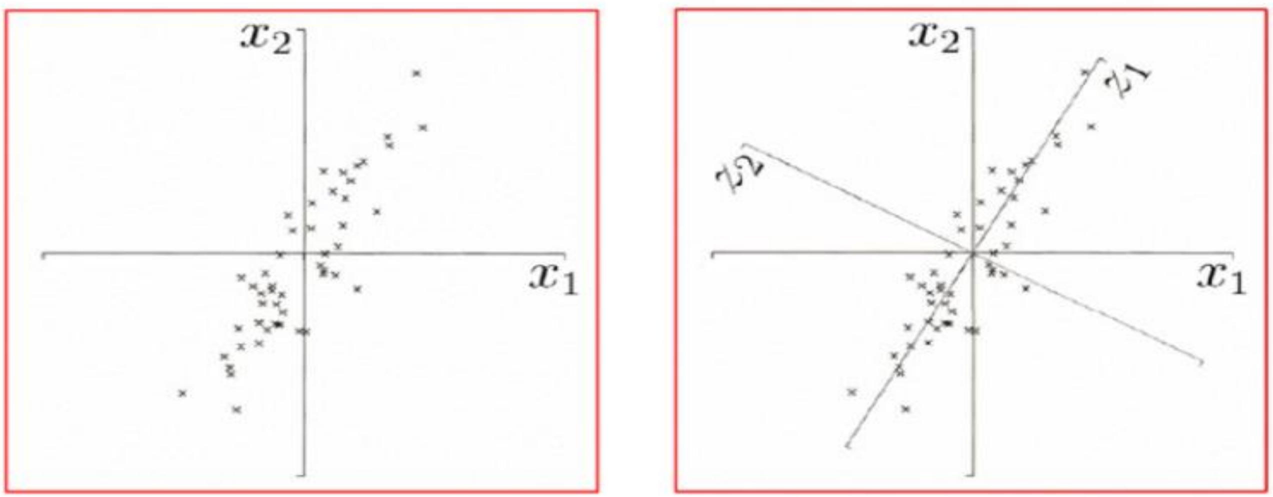
- Assume that the high dimensional data actually resides in a inherent low-dimensional space
- Additional dimensions are just random noise
- Goal is to recover these inherent dimensions and discard noise dimensions

Principal component analysis (PCA)

- Widely used method for unsupervised, linear dimensionality reduction
- GOAL: account for variance of data in as few dimensions as possible (using linear projection)

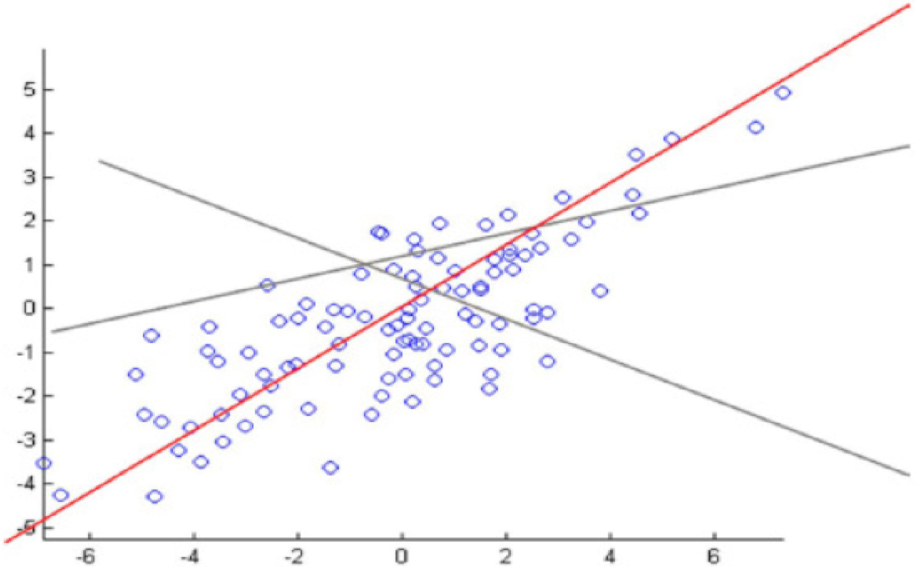
Geometric picture of principal components (PCs)

- First PC is the projection direction that maximizes the variance of the projected data
- Second PC is the projection direction that is orthogonal to the first PC and maximizes variance of the projected data



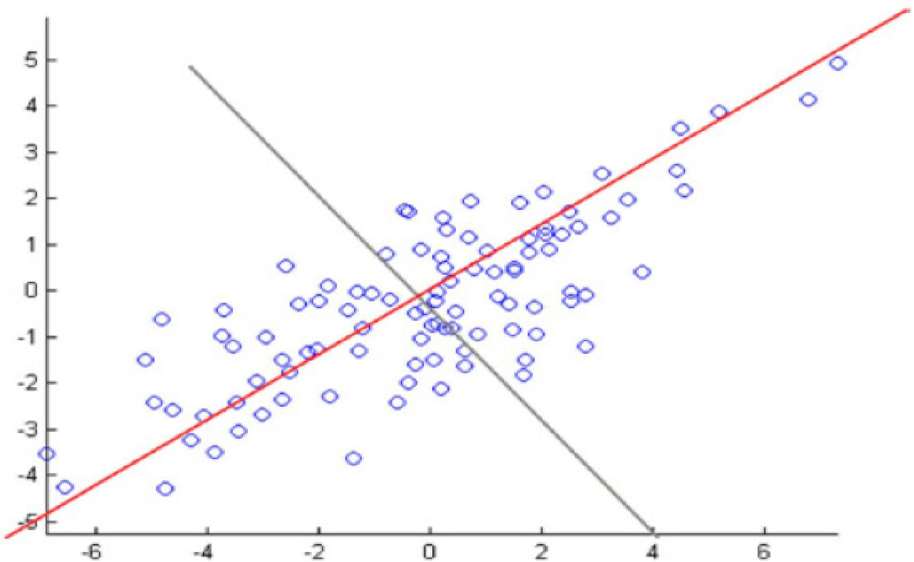
PCA: conceptual algorithm

- Find a line, such that when the data is projected onto that line, it has the maximum variance.



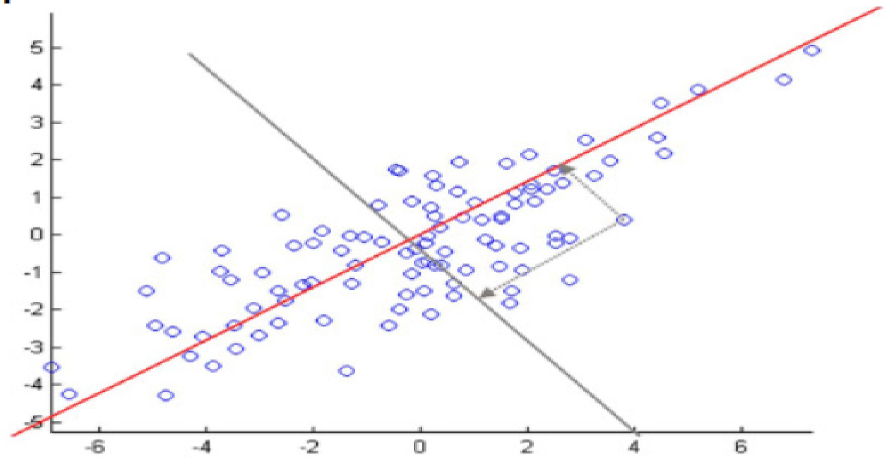
PCA: conceptual algorithm

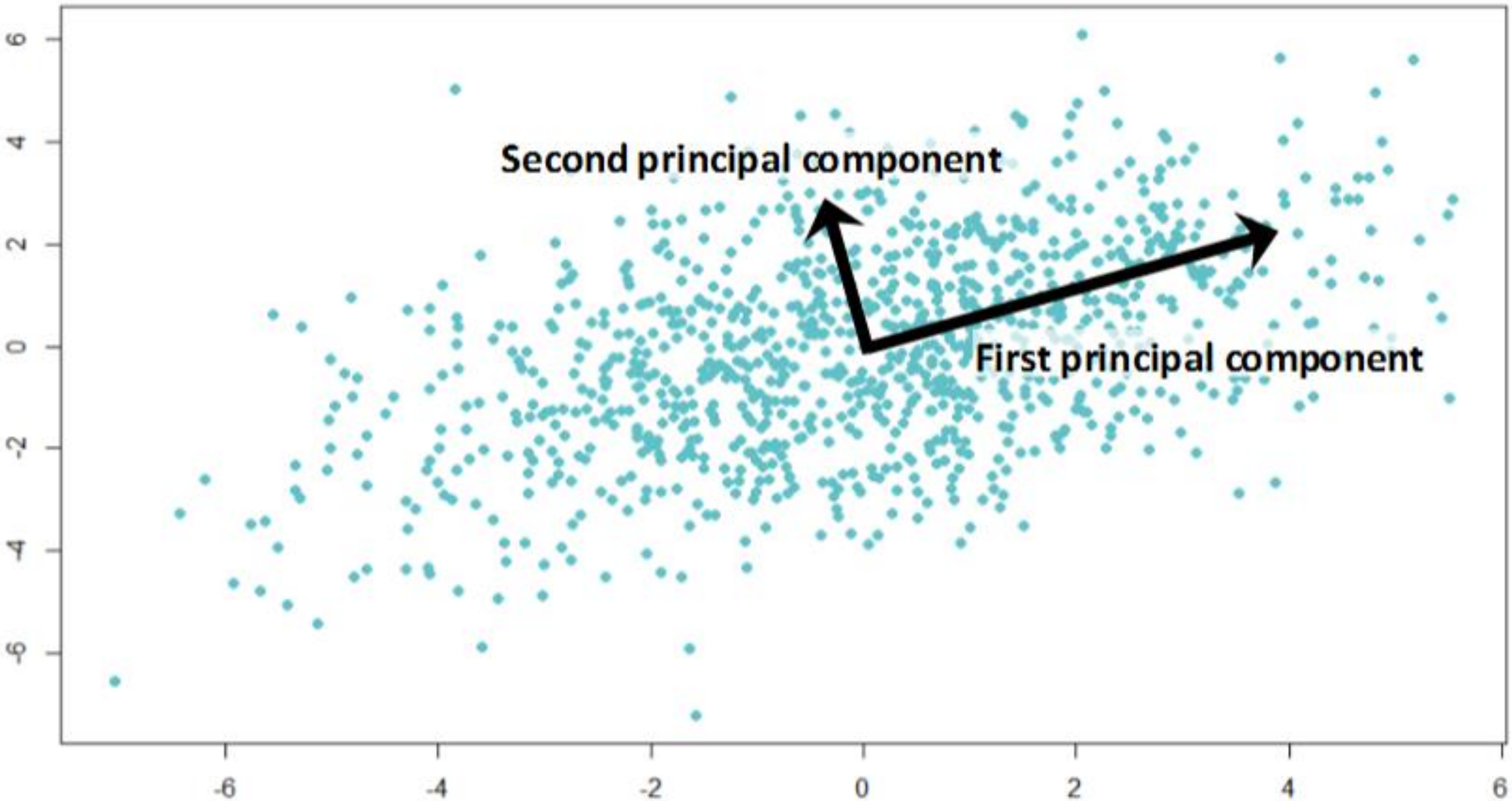
- Find a second line, orthogonal to the first, that has maximum projected variance.



PCA: conceptual algorithm

- Repeat until have *k* orthogonal lines
- The projected position of a point on these lines gives the coordinates in the *k*-dimensional reduced space.




Second principal component
First principal component
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

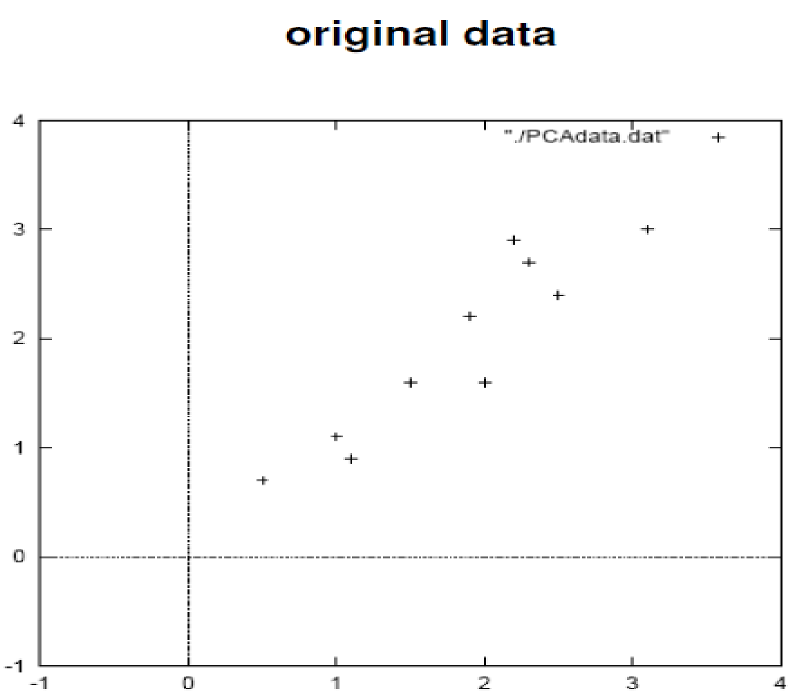
Steps in principal component analysis

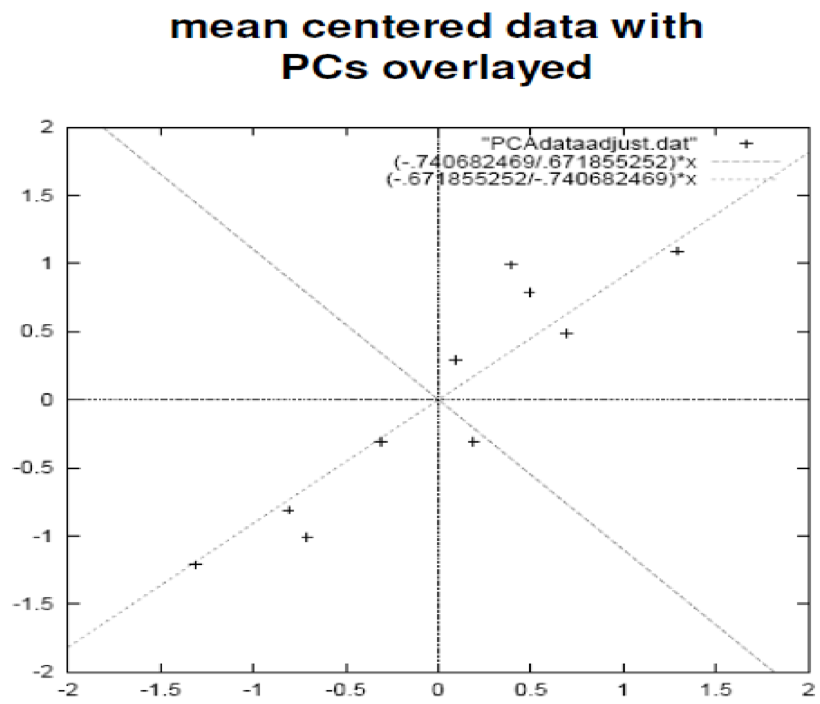
- Mean center the data
- Compute covariance matrix Σ
- Calculate eigenvalues and eigenvectors of Σ
  - Eigenvector with largest eigenvalue λ_1 is 1st principal component (PC)
  - Eigenvector with k^{th} largest eigenvalue λ_k is k^{th} PC
  - $\lambda_k / \Sigma_i \lambda_i$ = proportion of variance captured by k^{th} PC

Applying a principal component analysis

- Full set of PCs comprise a new orthogonal basis for feature space, whose axes are aligned with the maximum variances of original data.
- Projection of original data onto first *k* PCs gives a reduced dimensionality representation of the data.
- Transforming reduced dimensionality projection back into original space gives a reduced dimensionality *reconstruction* of the original data.
- Reconstruction will have some error, but it can be small and often is acceptable given the other benefits of dimensionality reduction.

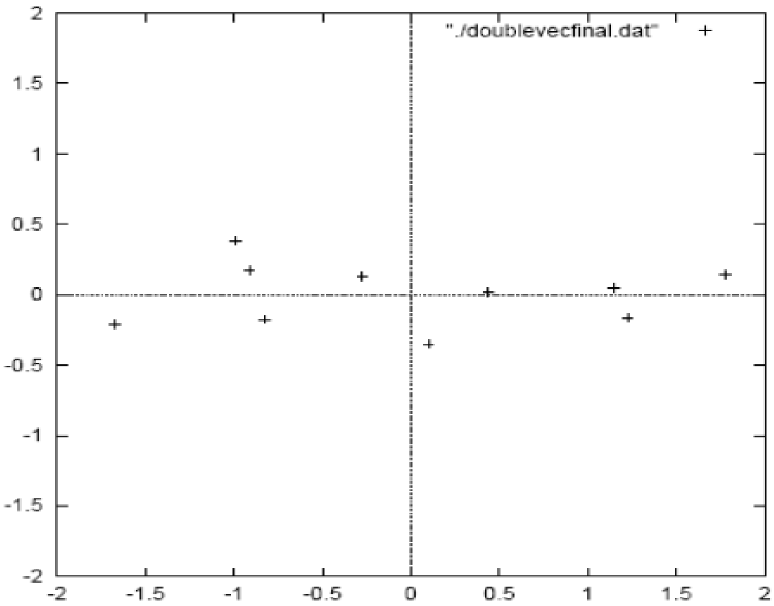
PCA example (1)



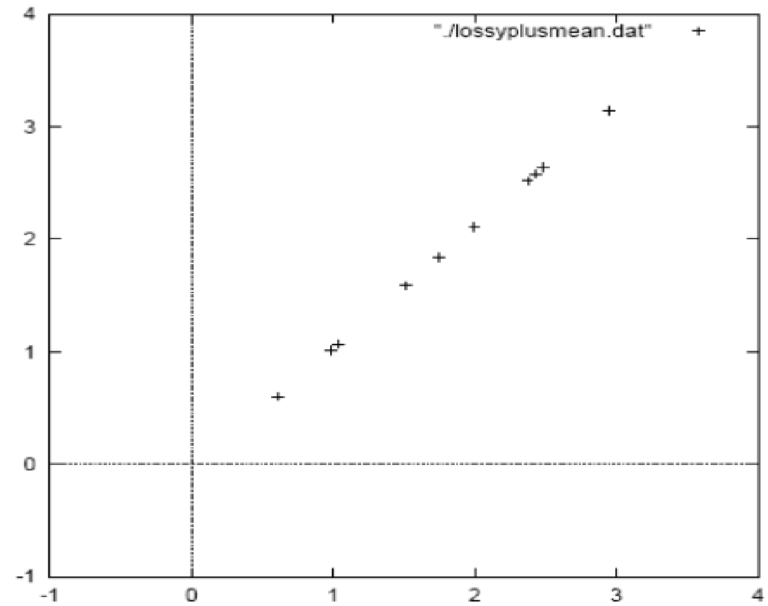


PCA example (1)

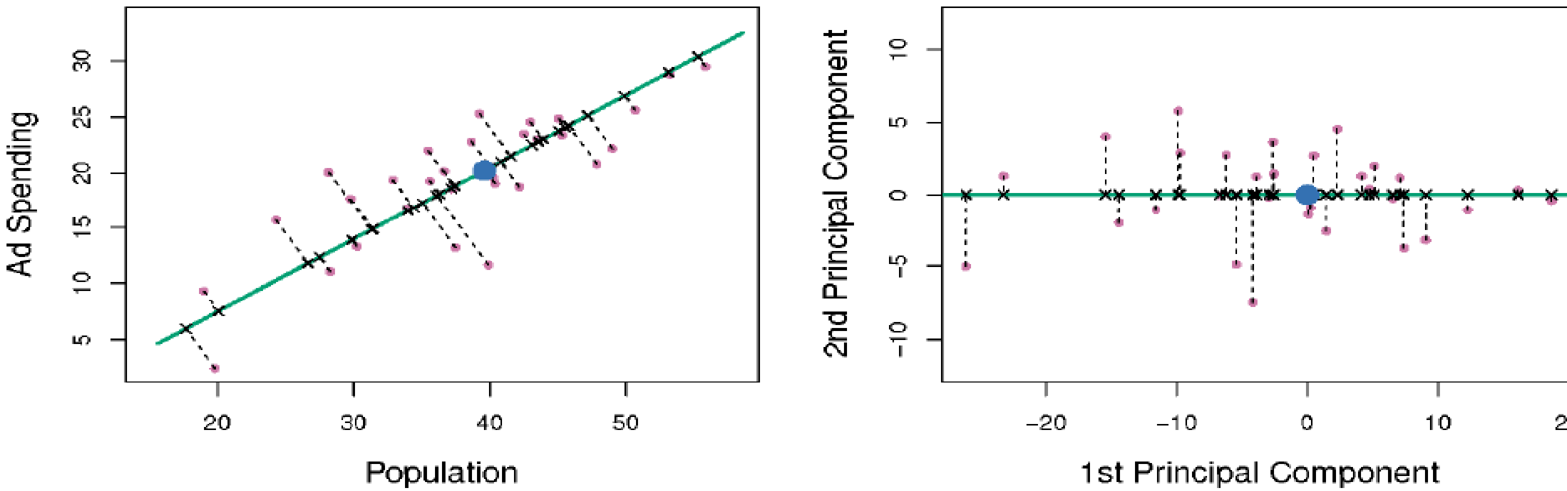
original data projected
Into full PC space



original data reconstructed using
only a single PC



PCA example (2)



Principal Component Analysis
(PCA)

Introduction

Principal component analysis (PCA) is a standard tool in modern data analysis - in diverse fields from neuroscience to computer graphics.

It is very useful method for extracting relevant information from confusing data sets.

Definition

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

The number of principal components is less than or equal to the number of original variables.

Goals

- The main goal of a PCA analysis is to identify patterns in data
- PCA aims to detect the correlation between variables.
- It attempts to reduce the dimensionality.

Dimensionality Reduction

It reduces the dimensions of a d-dimensional dataset by projecting it onto a (k)-dimensional subspace (where k<d) in order to increase the computational efficiency while retaining most of the information.

Transformation

This transformation is defined in such a way that the first principal component has the largest possible variance and each succeeding component in turn has the next highest possible variance.

PCA Approach

- Standardize the data.
- Perform Singular Vector Decomposition to get the Eigenvectors and Eigenvalues.
- Sort eigenvalues in descending order and choose the k-eigenvectors
- Construct the projection matrix from the selected k-eigenvectors.
- Transform the original dataset via projection matrix to obtain a k-dimensional feature subspace.

Limitation of PCA

The results of PCA depend on the scaling of the variables.

A scale-invariant form of PCA has been developed.

Applications of PCA :

- Interest Rate Derivatives Portfolios
- Neuroscience