Week 4

Course. Introduction to Machine Learning

Theory 4. Factor Analysis

Dr. Maria Salamó Llorente

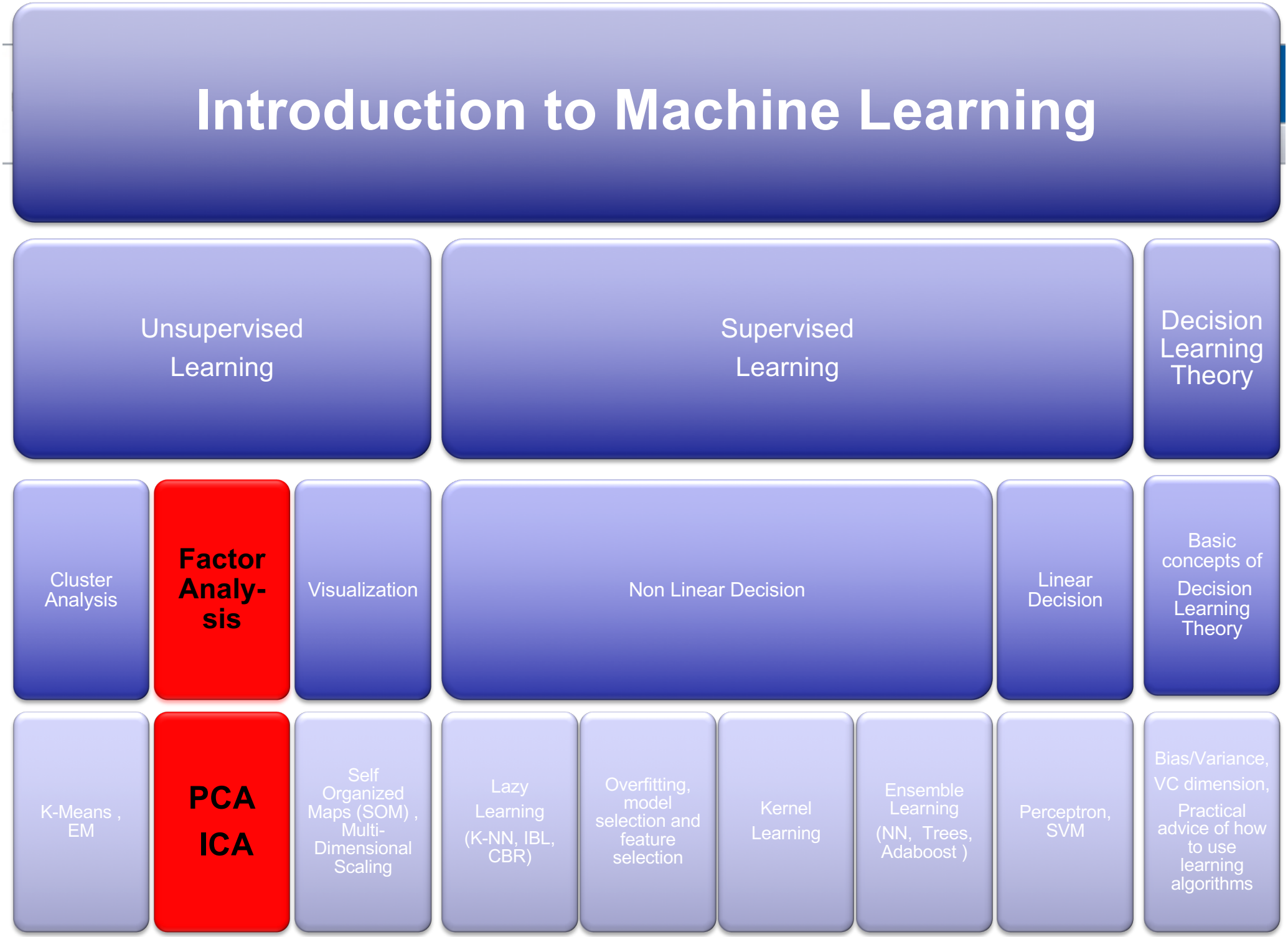
maria.salamo@ub.edu

Dept. Mathematics and Informatics,

Faculty of Mathematics and Informatics,

University of Barcelona (UB)

Introduction to Machine Learning



UNIVERSITAT DE BARCELONA

Contents

1. Introduction to Factor Analysis
2. Principal Component Analysis (PCA)
3. Independent Component Analysis (ICA)

Introduction to Factor Analysis

Introduction to Factor Analysis

He was a layer, a mathematician and statistician
(primary the latter)

Karl Pearson, 1857-1936

Inventor of the correlation coefficient, the chi-square test and Principal Components Analysis (1901)

UNIVERSITAT DE BARCELONA

Introduction to Factor Analysis

"Perhaps the most widely used (and misused) multivariate [technique] is factor analysis. Few statisticians are neutral about this technique. Proponents feel that factor analysis is the greatest invention since the double bed, while its detractors feel it is a useless procedure that can be used to support nearly any desired interpretation of the data. ***The truth, as is usually the case, lies somewhere in between****. Used properly, factor analysis can yield much useful information; when applied blindly, without regard for its limitations, it is about as useful and informative as Tarot cards.* ***In particular, factor analysis can be used to explore the data for patterns, confirm our hypotheses, or reduce the many variables to a more manageable number.***

-- Norman Streiner, PDQ Statistics

UNIVERSITAT DE BARCELONA

Introduction to Factor Analysis

Factor analysis (FA) is a method with some controversy as there are people that love it and there are others that consider it useless

USE OF FACTOR ANALYSIS:

- **To explore data for patterns**
- **Reduce the number of variables**

UNIVERSITAT DE BARCELONA

Introduction to Factor Analysis

- Factor analysis is a statistical method used to describe variability among **observed variables**, correlated variables in terms of a potentially lower number of **unobserved variables** called factors
  - In other words, it is possible, for example, that variations in three or four observed variables mainly reflect the variations in fewer unobserved variables
- Factor analysis searches for such joint variations in response to unobserved latent variables
- The observed variables are modeled as **linear combinations of the potential factors**, plus error terms

UNIVERSITAT DE BARCELONA

What is factor analysis?

GOAL: **To summarize patterns of correlations among observed variables**

- Factor analysis is a method for investigating whether a number of variables of interest are **linearly related** to a smaller number of unobservable factors
- Patterns of **correlations** are identified and either used as descriptive (PCA) or as indicative of underlying theory (FA)
- Process of providing an operational definition for **latent construct** (through a regression equation)

Latent variables

- Latent variables (as opposed to observable variables) are not directly observed but are rather inferred from other variables

Example:

- Quality of life is a latent variable
- Observable variables are: wealth, employment, environment, health, education, leisure time, and social belonging

UNIVERSITAT DE BARCELONA

Introduction to factor analysis

- FA (factor analysis) and PCA (principal components analysis) are statistical methods of **data reduction**
  - Take many **variables** and explain them with a few "factors" or "components"
  - **Correlated variables** are grouped together and separated from other variables with low or no correlation
  - Factors are formed that are relatively **independent** of one another.
  - Two types of "variables":
    - Latent variables: not directly observed but inferred
    - Observable variables: observed and directly measured

Dimensionality Reduction

- Factor analysis is an approach to deal with high dimensional data

- Project high dimensional data onto a lower dimensional sub-space using linear or non-linear transformations.

$\boldsymbol{x}=(x_1, x_2,\ldots, x_d)$ → $\boldsymbol{z}=(z_1, z_2,\ldots, z_k)$

Reduce dimensionality

$k << d$

UNIVERSITAT DE BARCELONA

General Steps to Factor Analysis

- **Step 1**: Selecting and measuring a set of variables in a given domain
- **Step 2**: Data screening in order to prepare the correlation matrix to perform PCA or FA
- **Step 3**: Factor Extraction
- **Step 4**: Factor Rotation to increase interpretability
- **Step 5**: Interpretation
- **Further Steps**: Validation and Reliability of the measures

What is a “Good Factor”?

- We consider a good factor when:
  - Makes sense
  - Will be easy to interpret
  - Simple structure
  - Lacks complex loadings

UNIVERSITAT DE BARCELONA

Problems with Factor Analysis

1. Unlike many of the analyses so far **there is no statistical criterion to compare the linear combination** to
   - In MANOVA, they create linear combinations that maximally differentiate groups
   - In Canonical correlation one linear combination is used to correlate with another
2. After extraction, there is an **infinite number of rotations** available
3. FA is frequently used to **"save" poorly conceived research**

UNIVERSITAT DE BARCELONA

Types of Factor Analysis

- **Exploratory** Factor Analysis
  - EFA is used to identify complex interrelationships among items and group items that are part of unified concepts
  - Summarizing data by grouping correlated variables
  - Investigating sets of measured variables related to theoretical constructs
  - Usually done near the onset of research

This is the type of FA and PCA we are talking about in this course

Types of Factor Analysis

- **Confirmatory** Factor Analysis
  - CFA is a **more complex approach** that tests the hypothesis that the items are associated with specific factors
  - When factor structure is known or at least theorized
  - Testing generalization of factor structure to new data, etc.

UNIVERSITAT DE BARCELONA

Summary of Factor Analysis

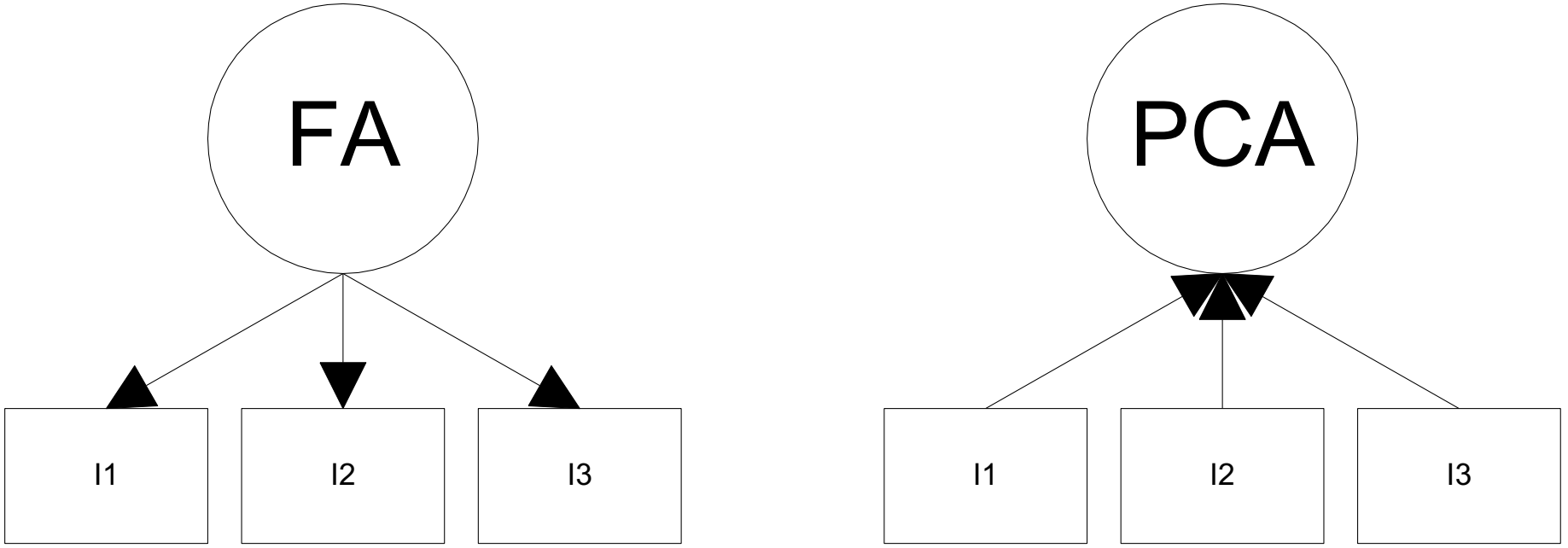
- Factor Analysis enables us to **reduce** the **complexity of data**
- The exercise of data reduction makes it possible to identify the **latent variables** which exist underneath a set of variables which are actually observed
- These latent variables can then be used for further analysis to make sense of relationships in the data

Applications of Factor Analysis

1. Identification of Underlying Factors:
   - clusters variables into homogeneous sets
   - creates new variables (i.e. factors)
   - allows us to gain insight to categories
2. Screening of Variables:
   - identifies groupings to allow us to select one variable to represent many
   - useful in regression (recall collinearity)
3. Summary:
   - Allows us to describe many variables using a few factors
4. Clustering of objects:
   - Helps us to put objects (people) into categories depending on their factor scores

Principal Components Analysis (PCA)

FA vs. PCA



FA is used to understand what constructs underlie the data.
PCA is used to reduce the data into a smaller number of components.

UNIVERSITAT DE BARCELONA

FA vs. PCA

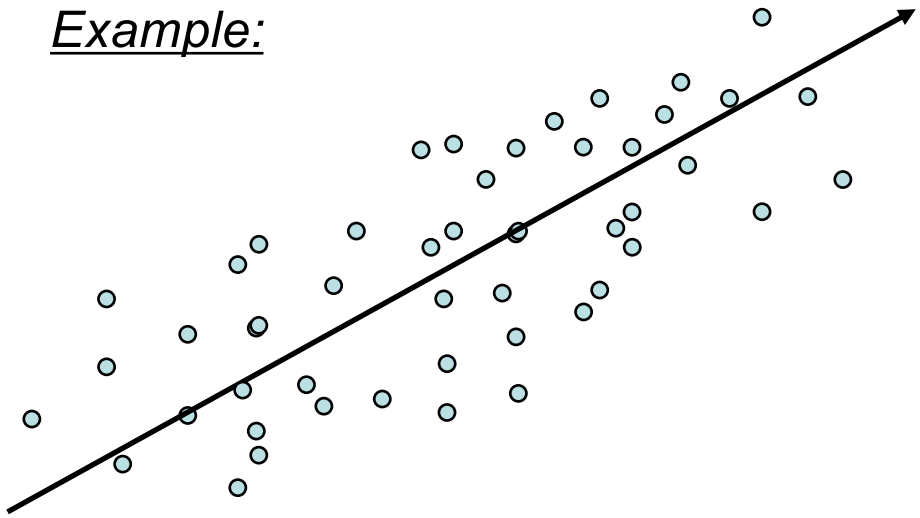
- Principal Components Analysis and Factor Analysis are similar because both procedures are used to simplify the structure of a set of variables. However, the analyses differ in several important ways:
  - In PCA, the components are calculated as linear combinations of the original variables
  - In FA, the original variables are defined as linear combinations of the factors
  - In PCA, the goal is to **account for as much of the total** variance in the variables as possible
  - The objective in FA **is to explain the** covariances or **correlations** among the variables

UNIVERSITAT DE BARCELONA

Principal Component Analysis

- Imagine that we have many examples of the same pattern.
- Data appear clouded, unclear and even redundant.

Example:



***Goal** of PCA: find new representation (basis) to filter the noise and reveal hidden dynamics*

Principal Component Analysis

- It can be seen as the 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

UNIVERSITAT DE BARCELONA

Principal Component Analysis

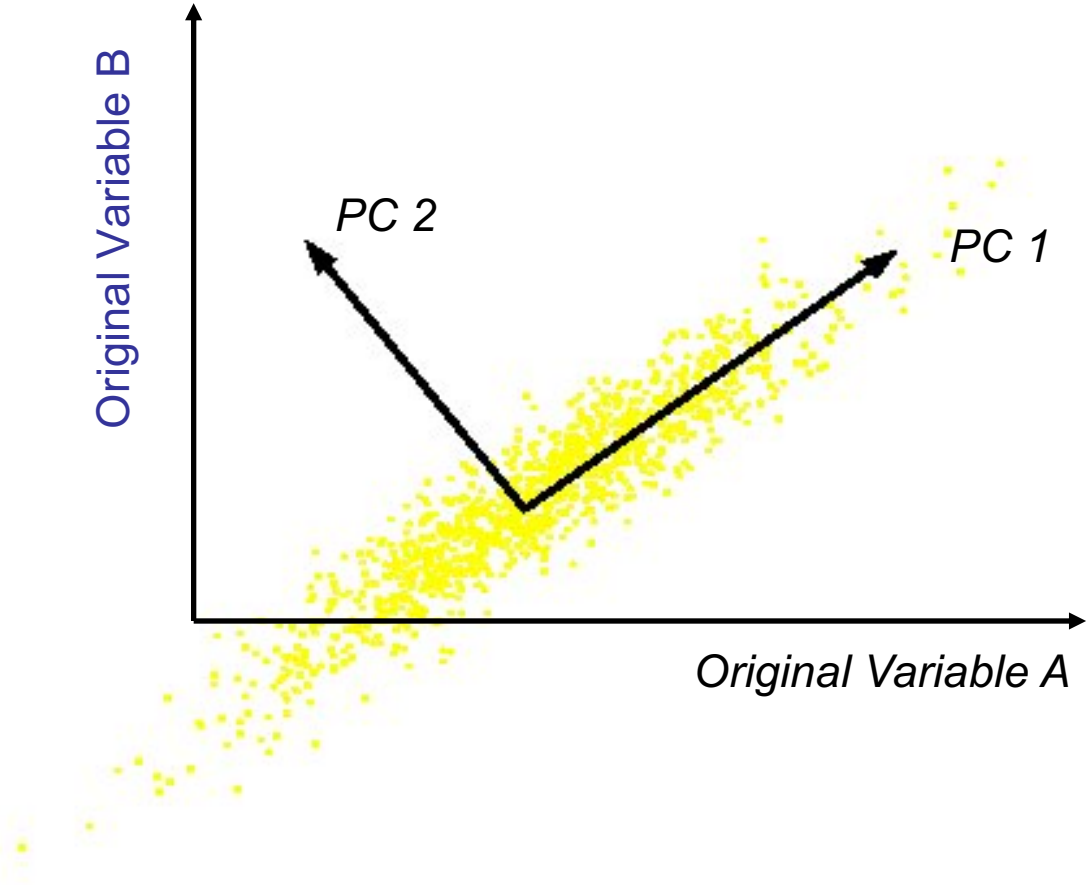
- Principal component analysis (PCA) involves a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called *principal components*
- The first principal component accounts for the greatest possible statistical variability (or entropy) in the data,
  - And each succeeding component accounts for as much of the remaining variability as possible
- Usually, PCA is used to discover or **reduce the dimensionality** of the data set, or to identify new meaningful underlying variables, i.e., patterns in the data.

What PCA does?

- PCA is "**an orthogonal linear transformation** that transfers the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (*first principal component*), the second greatest variance lies on the second coordinate (*second principal component*), and so on"
  - Find new axes
  - Decide on which are significant
  - Form a new coordinate system defined by the significant axes
  - Map data to the new space (Compressed Data)

UNIVERSITAT DE BARCELONA

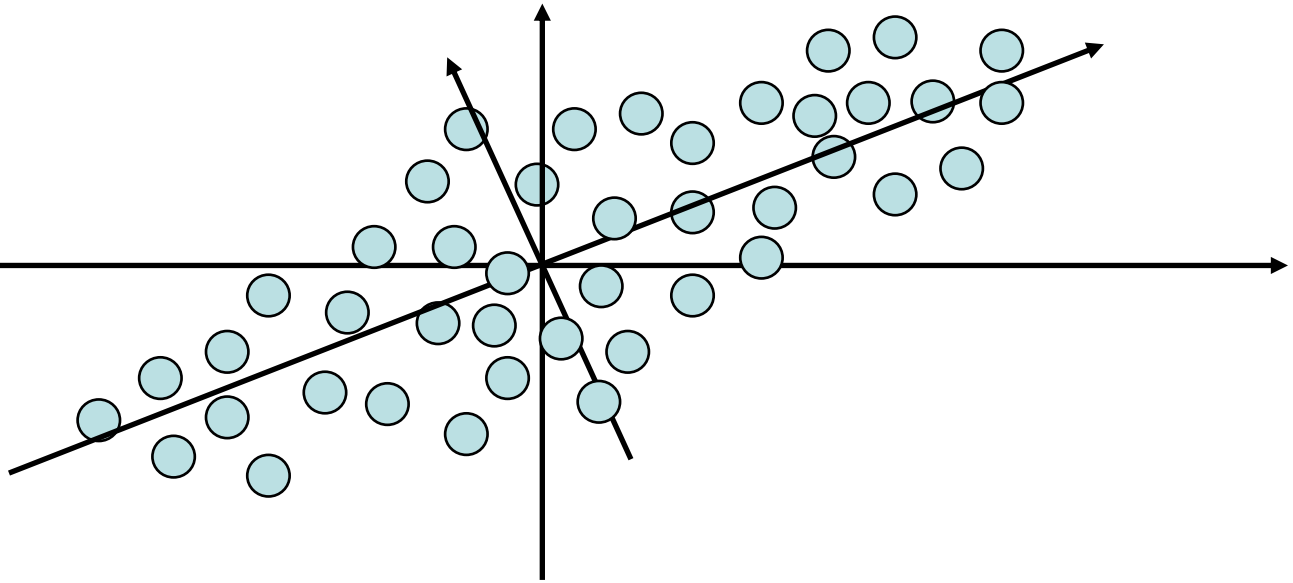
What PCA does?



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

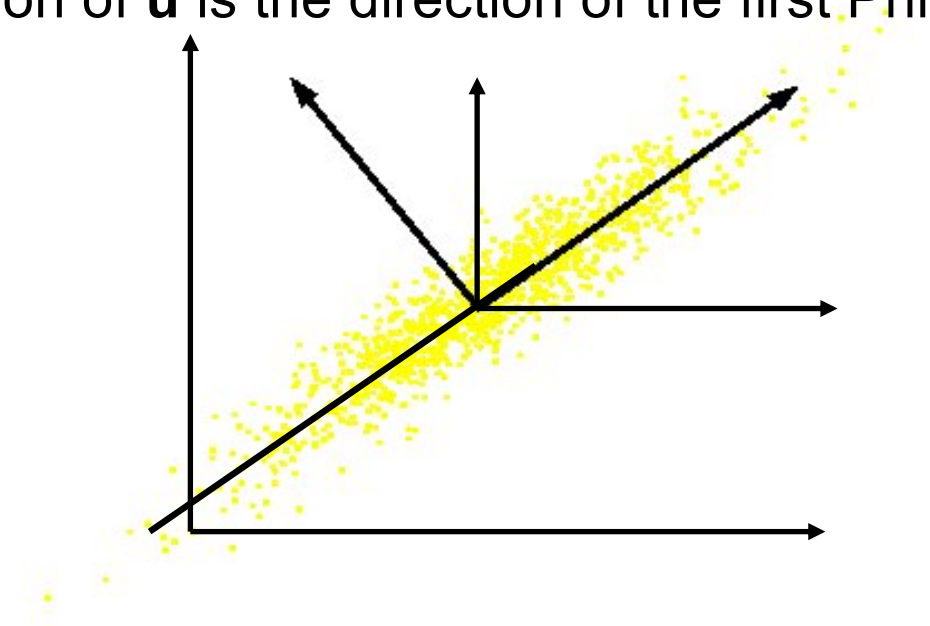
Principal Components

- **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 **u.x**
- Direction of greatest variability is that in which the average square of the projection is greatest
  - I.e. **u** such that $E((\mathbf{u}.\mathbf{x})^2)$ over all **x** is maximized
  - (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



UNIVERSITAT DE BARCELONA

PCA in a nutshell

- 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

Geometric Rationale of PCA

- Objects are represented as a cloud of *n* points in a multidimensional space with an axis for each of the *p* variables
  - First, the centroid of the points is defined by the mean of each variable
  - Second, the variance of each variable is the average squared deviation of its *n* values around the mean of that variable.

$$V_i = \frac{1}{n-1}\sum_{m=1}^{n}\left(X_{im} - \overline{X}_i\right)^2$$

Background for PCA

- Suppose:
  - attributes are A_1 and A_2,
  - we have *n* training examples. *x*'s denote values of A_1 and *y*'s denote values of A_2 over the training examples.

- Variance of an attribute:

$$\text{var}(A_1) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{(n-1)}$$

Background for PCA

- Covariance of two attributes:

$$\text{cov}(A_1, A_2) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{(n-1)}$$

- A **positive** covariance means both dimensions increase together.
- A **negative** covariance means as one dimension increases, the other decreases.
- A **Zero** covariance means dimensions are independent of each other.

Covariance matrix

- Covariance matrix
  - Suppose we have ***n*** attributes, $A_1, ..., A_n$.
  - Covariance matrix:

$$C^{n\times n} = (c_{i,j}), \text{where } c_{i,j} = \text{cov}(A_i, A_j)$$

$$\begin{pmatrix} \text{cov}(H,H) & \text{cov}(H,M) \\ \text{cov}(M,H) & \text{cov}(M,M) \end{pmatrix}$$

	Hours(H)	*Mark(M)*
Data	9	39
	15	56
	25	93
	14	61
	10	50
	18	75
	0	32
	16	85
	5	42
	19	70
	16	66
	20	80
Totals	167	749
Averages	13.92	62.42

Covariance matrix

$$\begin{pmatrix} \text{cov}(H,H) & \text{cov}(H,M) \\ \text{cov}(M,H) & \text{cov}(M,M) \end{pmatrix}$$

H	M	$(H_i - \bar{H})$	$(M_i - \bar{M})$	$(H_i - \bar{H})(M_i - \bar{M})$
9	39	-4.92	-23.42	115.23
15	56	1.08	-6.42	-6.93
25	93	11.08	30.58	338.83
14	61	0.08	-1.42	-0.11
10	50	-3.92	-12.42	48.69
18	75	4.08	12.58	51.33
0	32	-13.92	-30.42	423.45
16	85	2.08	22.58	46.97
5	42	-8.92	-20.42	182.15
19	70	5.08	7.58	38.51
16	66	2.08	3.58	7.45
20	80	6.08	17.58	106.89
Total				1149.89
Average				104.54

$$= \begin{pmatrix} \text{var}(H) & 104.5 \\ 104.5 & \text{var}(M) \end{pmatrix}$$

$$= \begin{pmatrix} 47.7 & 104.5 \\ 104.5 & 370 \end{pmatrix}$$

Covariance matrix

Eigenvectors

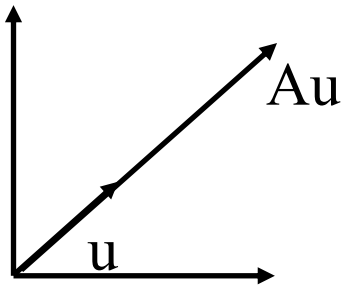
- An eigenvector is a vector $\boldsymbol{v}$ that obeys the following rule:

$$\mathrm{Au} = \lambda\,\mathrm{u}$$

where A is a matrix, λ is a scalar (called the eigenvalue)

Example: $\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}$ one eigenvector of A is $\mathrm{u} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$ since

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \end{bmatrix} = 4 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$



→ so for this eigenvector of A the eigenvalue is 4.

UNIVERSITAT DE BARCELONA

Eigenvectors

- We can think of matrices as performing transformations on vectors (e.g. rotations, reflections)
- We can think of eigenvectors of a matrix as being special vectors (for that matrix) that are **only** scaled by that matrix
- Different matrices have different eigenvectors
- **Only square matrices have eigenvectors**
- **Not all square matrices have eigenvectors**
- An $n \times n$ matrix has **at most** n distinct eigenvectors
- All the eigenvectors of a $n \times n$ matrix, with n different eigenvalues, are *perpendicular (or orthogonal)*

Eigenvectors

- Let M be an *n*×*n* matrix.
  - v is an *eigenvector* of M if M × v = λv
  - λ is called the *eigen-value* associated with v
- For any eigenvector v of M and scalar *a*,

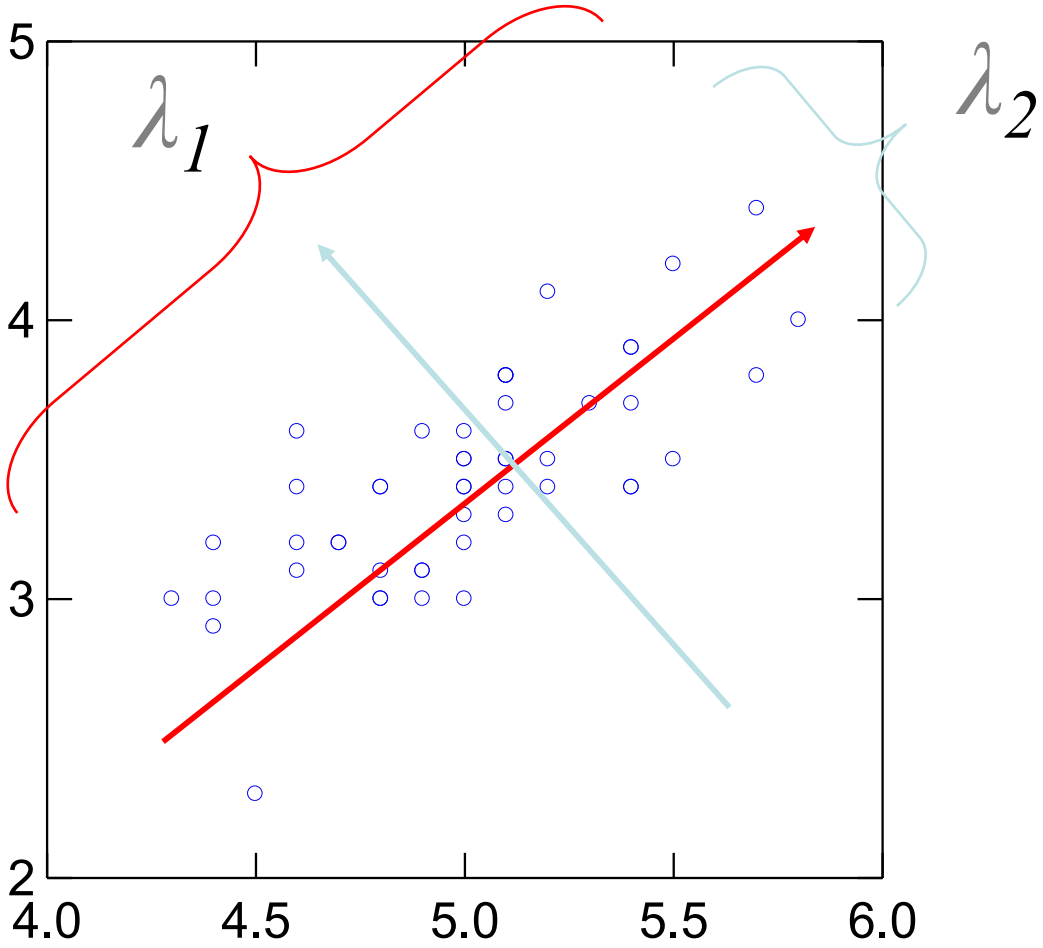
$$\mathbf{M} \times a\mathbf{v} = \lambda a\mathbf{v}$$

- Thus you can always choose eigenvectors of length 1:

$$\sqrt{v_1^2 + \ldots + v_n^2} = 1$$

- If M has any eigenvectors, it has **n** of them, and **they are orthogonal to one another**.
- **Thus eigenvectors can be used as a new basis for a n-dimensional vector space**.

PCA Eigenvalues



Steps for PCA

0. Get some data
1. **Subtract the mean** from data
2. Calculate the **covariance** matrix
3. Calculate the **eigenvectors** and eigenvalues of the covariance matrix
4. **Choose components** and construct a new feature vector
5. Derive the **new data set**
6. **Reconstruct** the old data back

PCA example

1. Given original data set S = $\{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$, produce new set by subtracting the mean of attribute A_i from each x_i.

Data =

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

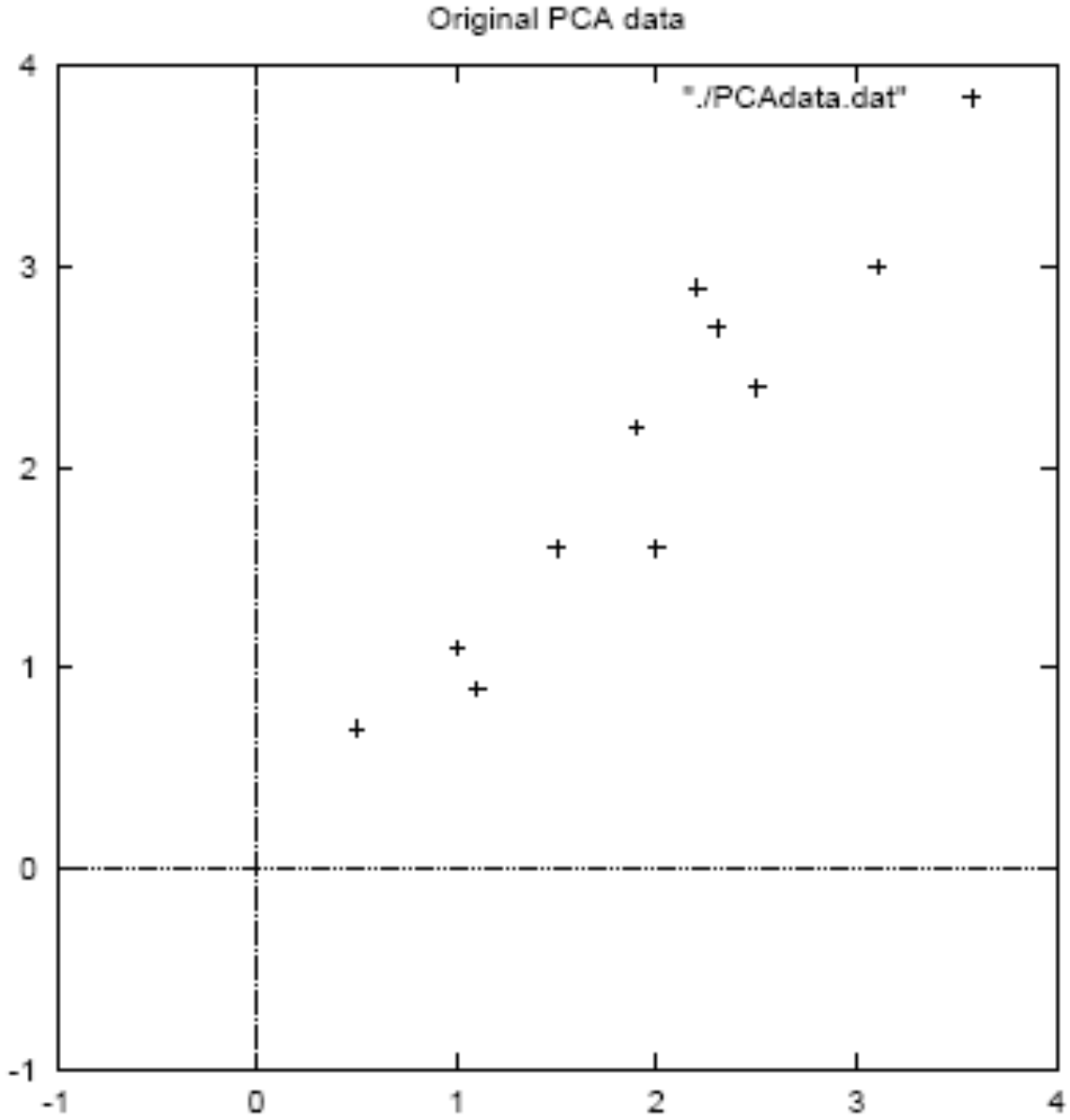
Mean: 1.81 1.91

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

Mean: 0 0

PCA example: original data set



PCA example

2. Calculate the **covariance** matrix:

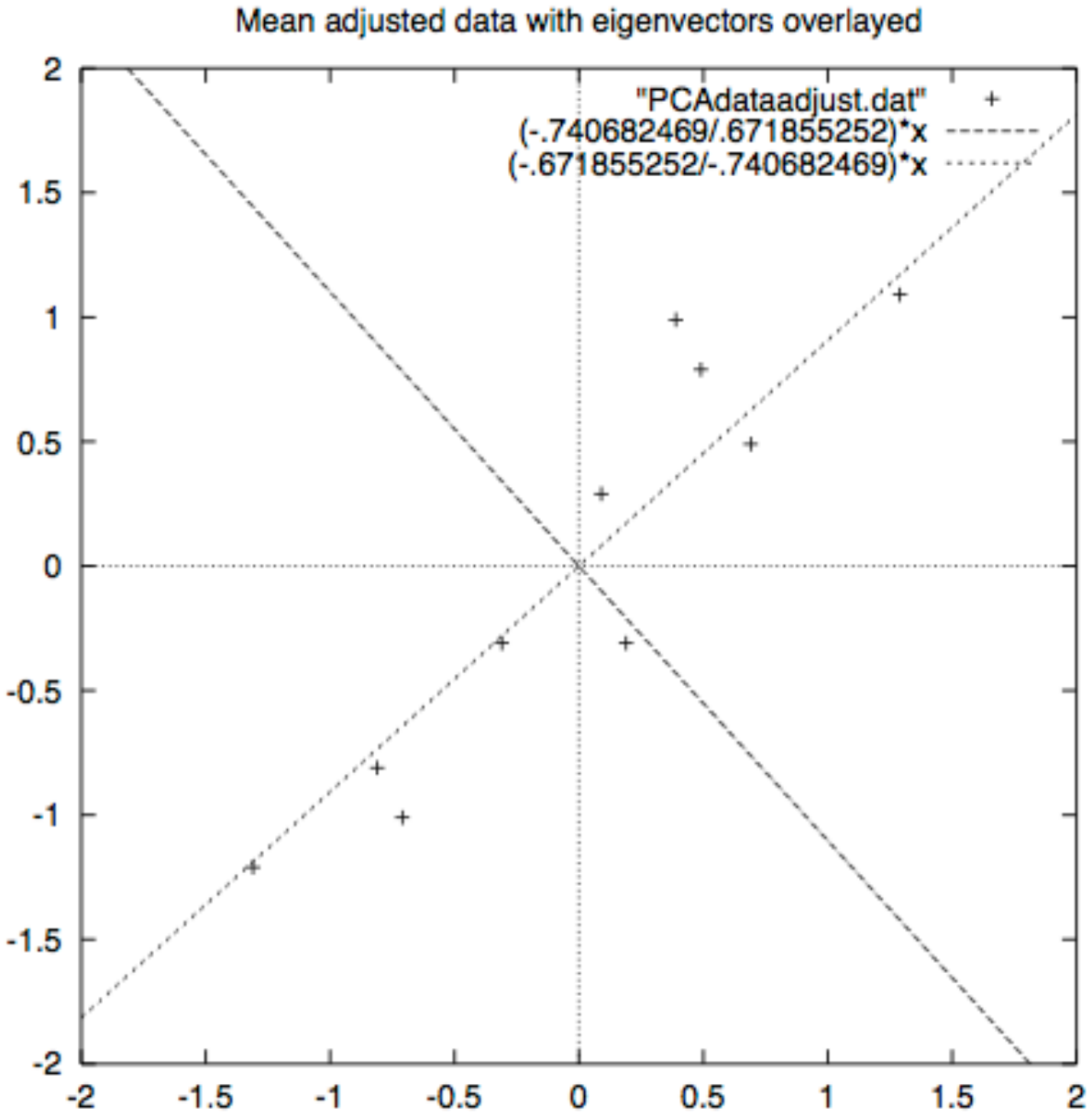
$$cov = \begin{array}{c} \\ x \\ y \end{array} \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix} \end{array}$$

3. Calculate the (unit) eigenvectors and **eigenvalues** of the covariance matrix:

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

PCA example



Eigenvector with largest eigenvalue traces linear pattern in data

PCA example

Order eigenvectors by eigenvalue, highest to lowest.

$$\mathbf{v}_1 = \begin{pmatrix} -.677873399 \\ -.735178956 \end{pmatrix} \quad \lambda = 1.28402771$$

$$\mathbf{v}_2 = \begin{pmatrix} -.735178956 \\ .677873399 \end{pmatrix} \quad \lambda = .0490833989$$

In general, you get *n* components.

To reduce dimensionality to *p*, ignore *n*–*p* components at the bottom of the list.

PCA example

4. Construct new feature vector.

Feature vector = ($\mathbf{v}_1$, $\mathbf{v}_2$, ...$\mathbf{v}_p$)

$$FeatureVector1 = \begin{pmatrix} -.677873399 & -.735178956 \\ -.735178956 & .677873399 \end{pmatrix}$$

or reduced dimension feature vector :

$$FeatureVector2 = \begin{pmatrix} -.677873399 \\ -.735178956 \end{pmatrix}$$

PCA example

5. Derive the new data set.

$$TransformedData = RowFeatureVector \times RowDataAdjust$$

$$RowFeatureVector1 = \begin{pmatrix} -.677873399 & -.735178956 \\ -.735178956 & .677873399 \end{pmatrix}$$

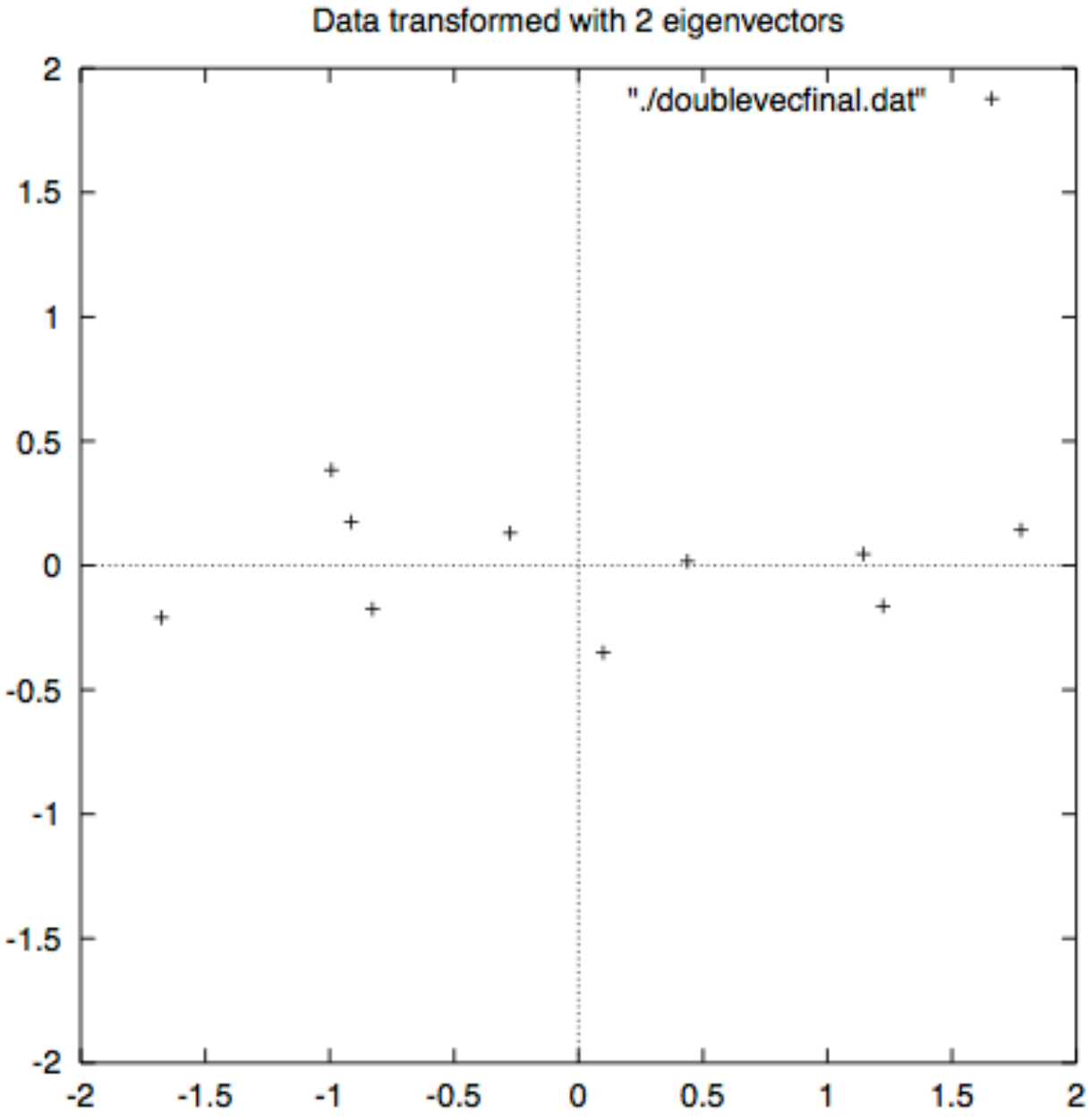
$$RowFeatureVector2 = \begin{pmatrix} -.677873399 & -.735178956 \end{pmatrix}$$

$$RowDataAdjust = \begin{pmatrix} .69 & -1.31 & .39 & .09 & 1.29 & .49 & .19 & -.81 & -.31 & -.71 \\ .49 & -1.21 & .99 & .29 & 1.09 & .79 & -.31 & -.81 & -.31 & -1.01 \end{pmatrix}$$

PCA example

Transformed data

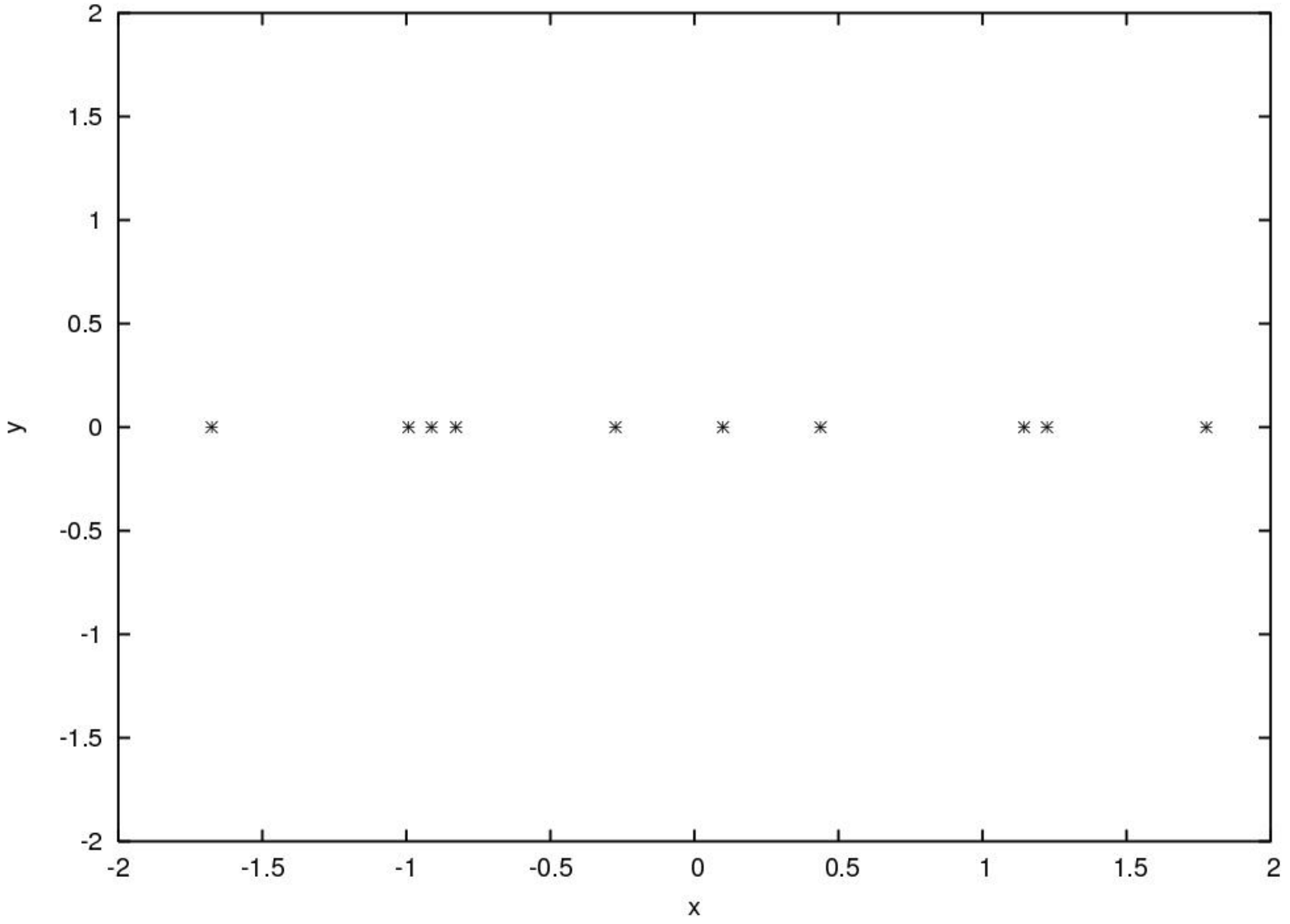
x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287



PCA example

Transformed data with a Single eigenvector

x
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056



PCA example

6. Reconstruct the old data back

- We did:

$$\mathbf{TransformedData = RowFeatureVector \times RowDataAdjust}$$

- so we can do

$$\mathbf{RowDataAdjust = RowFeatureVector\ -1 \times TransformedData}$$

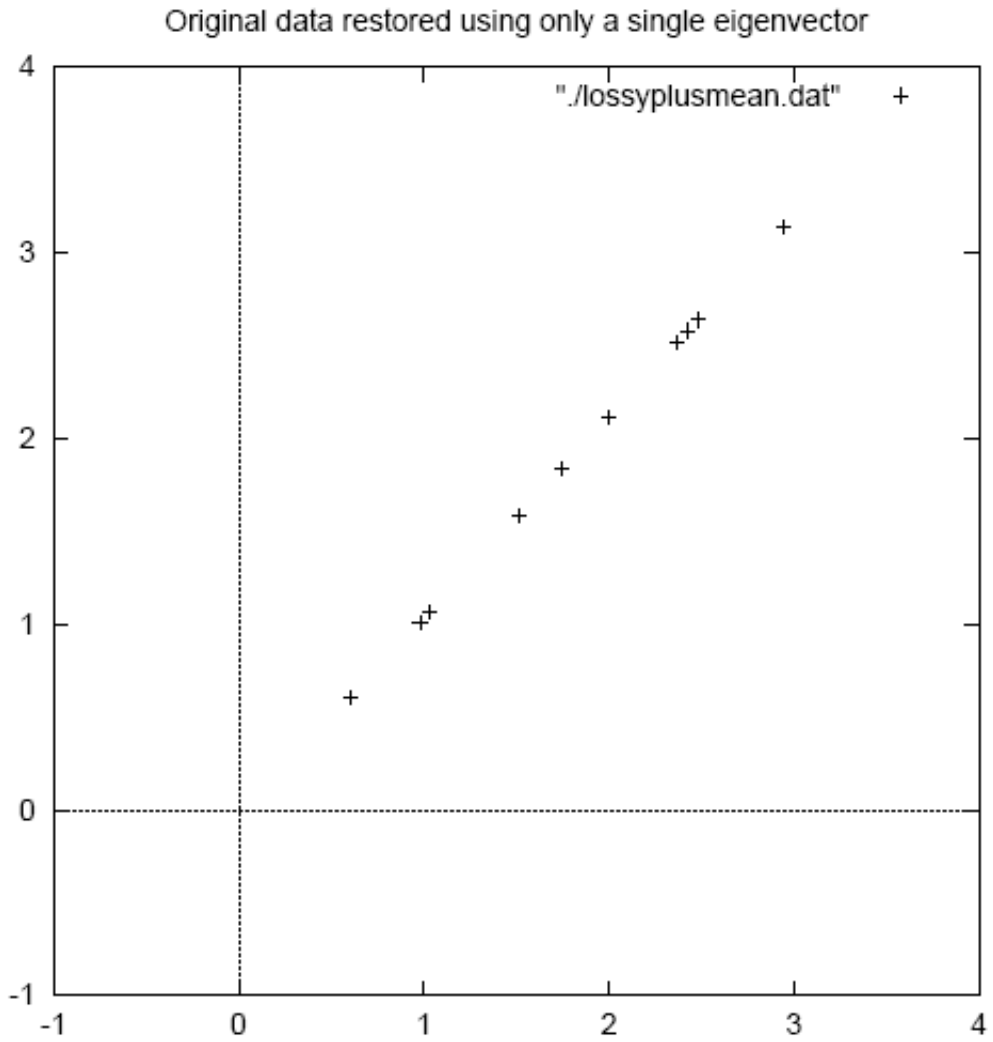
$$\mathbf{= RowFeatureVector\ T \times TransformedData}$$

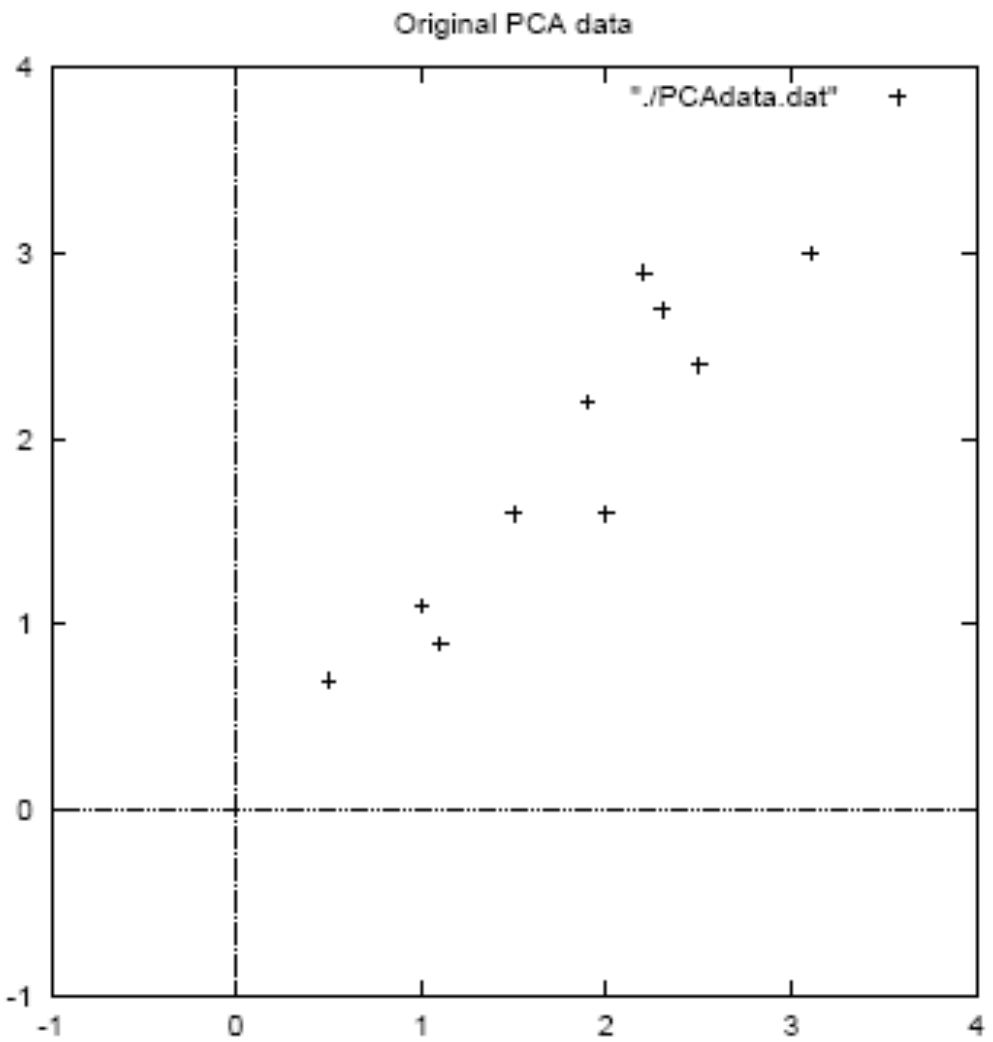
- and

$$\mathbf{RowDataOriginal = RowDataAdjust + OriginalMean}$$

UNIVERSITAT DE BARCELONA

PCA example



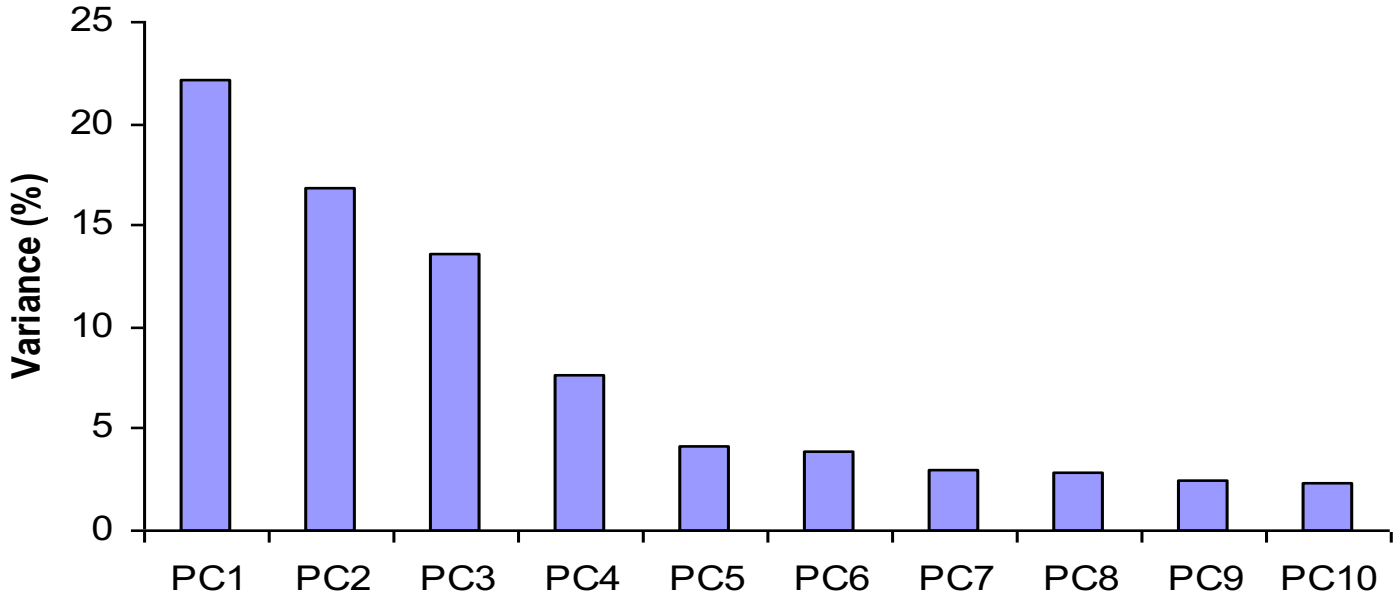


How Many PCs?

- For **n** original dimensions, correlation matrix is **n x n**, and has up to **n** eigenvectors. So **n** PCs.
- As a rule of thumb, it is worth the selection of those **eigenvalues > 1**
- Where does dimensionality reduction come from?

Dimensionality Reduction

Can ignore the components of lesser significance.



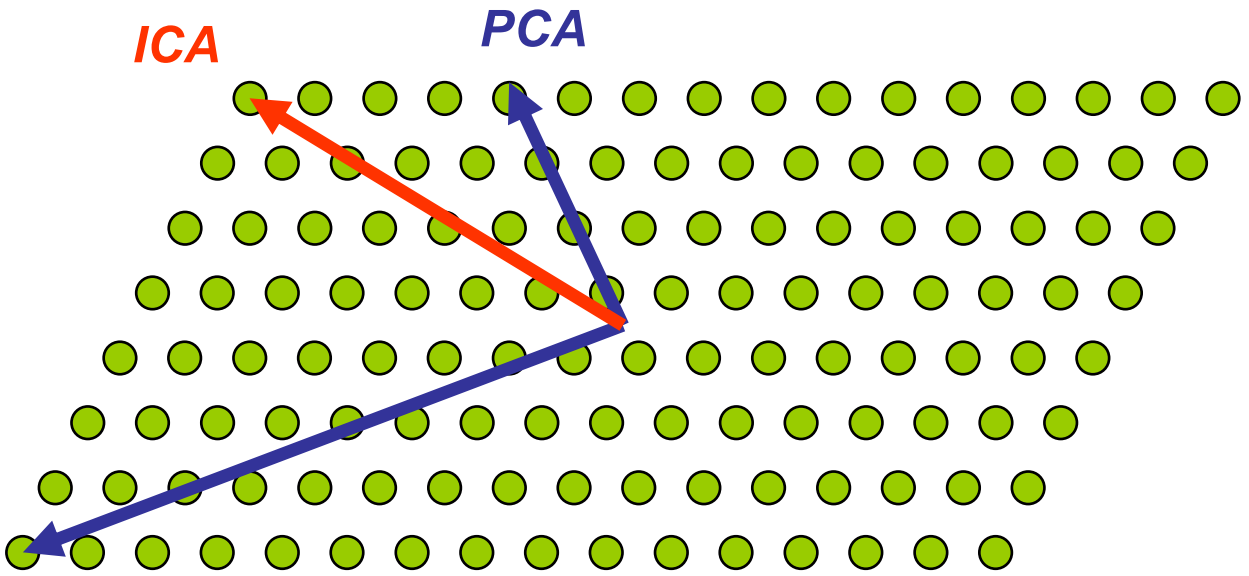
You do lose some information, but if the eigenvalues are small, you don't lose much

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

Limitations of PCA

Should the goal be finding independent rather than pair-wise uncorrelated dimensions

- ***Independent Component Analysis (ICA)***



UNIVERSITAT DE BARCELONA

Links

Look at the video

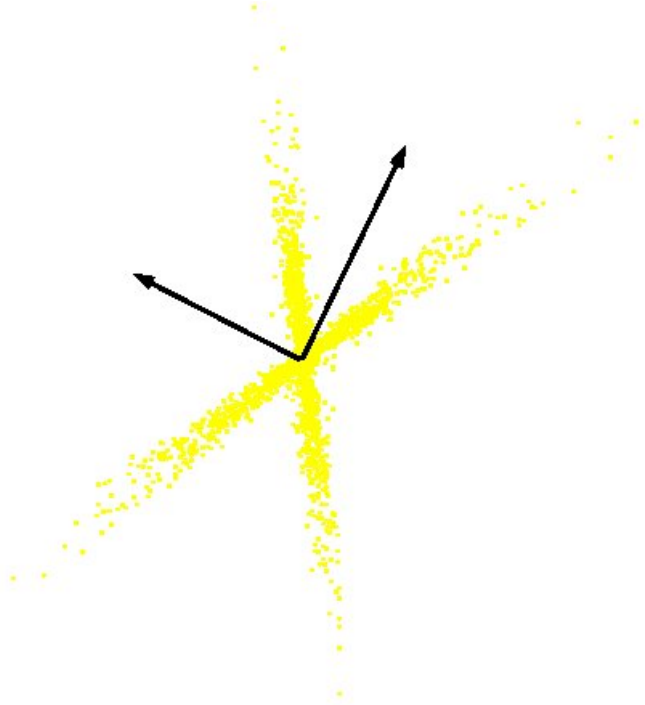
https://youtu.be/g-Hb26agBFg

TO READ: (OPTIONAL)

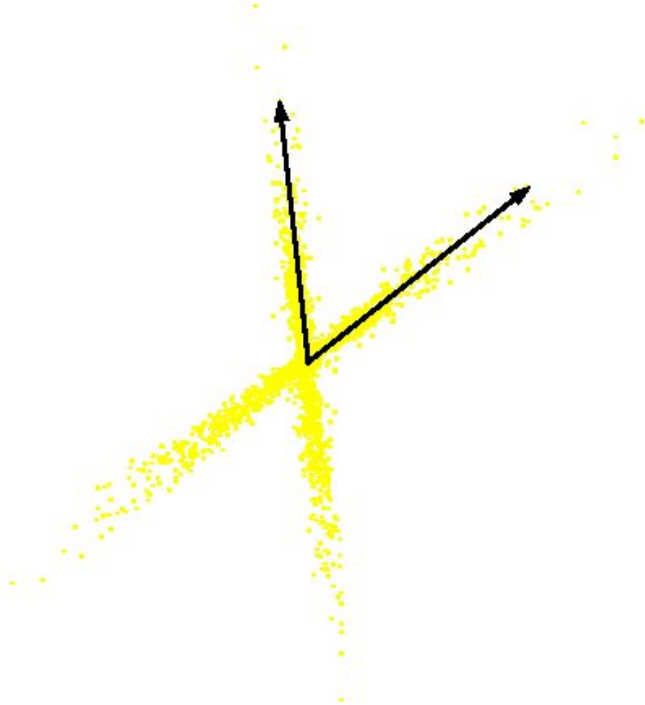
https://medium.com/@aptrishu/understanding-principle-component-analysis-e32be0253ef0

Independent Component Analysis (ICA)

PCA vs ICA



PCA
(orthogonal coordinate)



ICA
(non-orthogonal coordinate)

PCA vs ICA

- PCA
  - **Finds the directions of maximum variance**
  - Focus on uncorrelated and Gaussian components
  - Second-order statistics
  - Orthogonal transformation

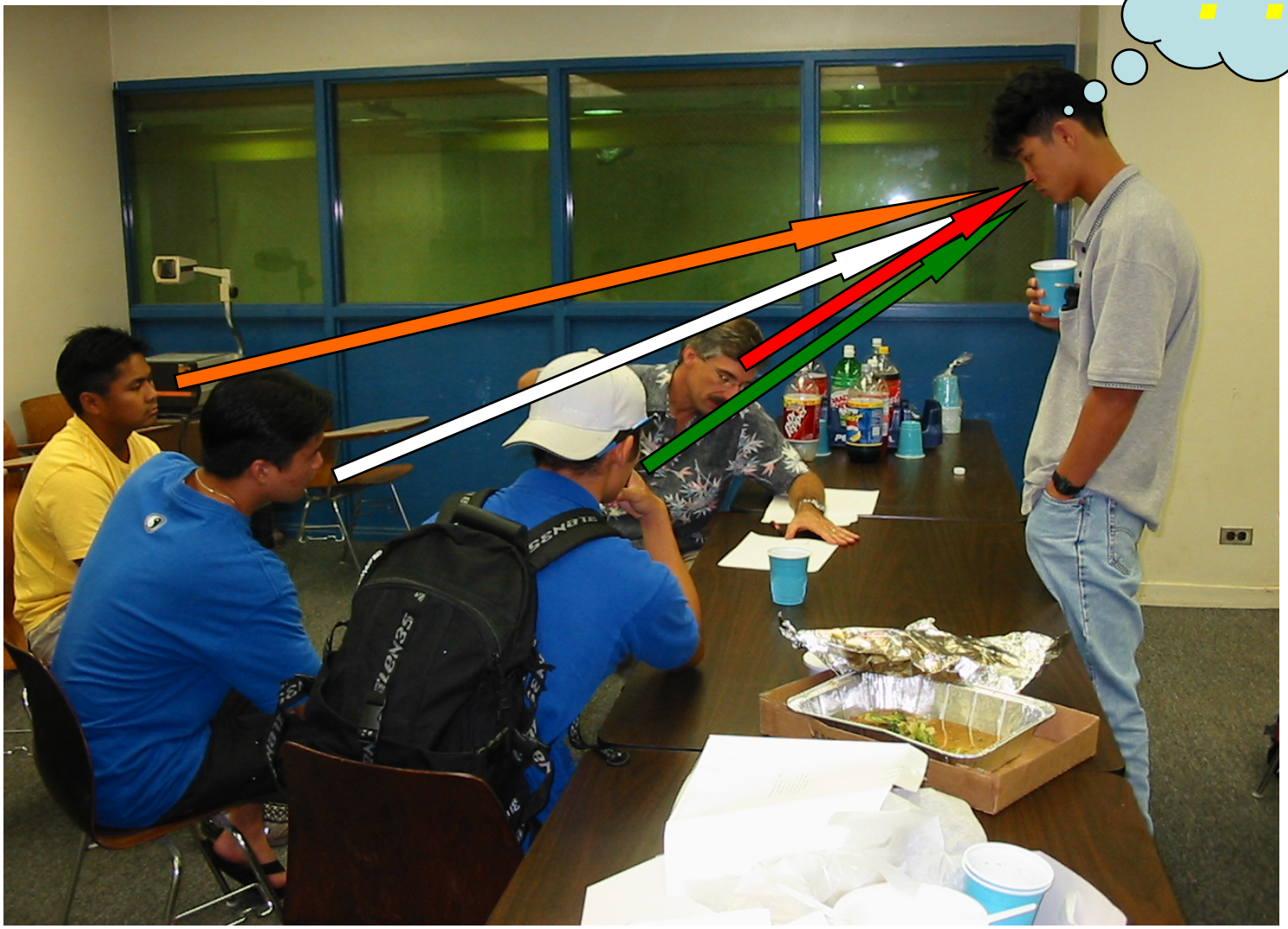
- ICA
  - **Finds the directions of maximum independence**
  - Focus on independent and non-Gaussian components
  - Higher-order statistics
  - Non-orthogonal transformation

Motivation for ICA

- Most measured quantities are actually mixtures of other quantities
  - Sound signals, EEG signals, Magnetic resonance, …
- **Cocktail Party Problem**
  - Suppose you are in a crowded room with many people. How do you understand what any one person is saying?
- **Separation of Independent Signals**
  - Similar to Blind Source Separation
  - Little knowledge of the signals
  - Access to mixed signals only

UNIVERSITAT DE BARCELONA

Cocktail Party Problem

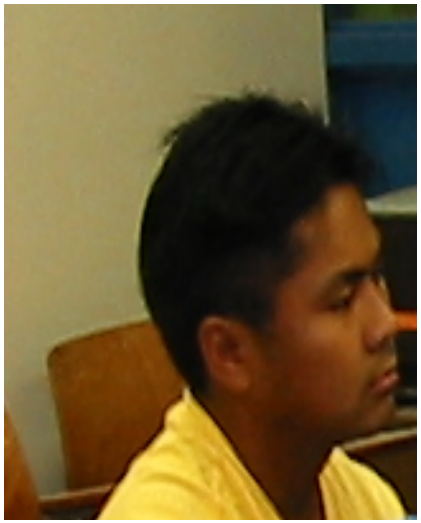


Cocktail Party Problem

- ICA Separation Algorithm
  - Separation of Speech Signals
  - Humans can separate multiple signals with only two ears/sensors
  - ICA needs as many ears/sensors as message signals
    - Here we assume he has four ears!

UNIVERSITAT DE BARCELONA

Recovered Messages

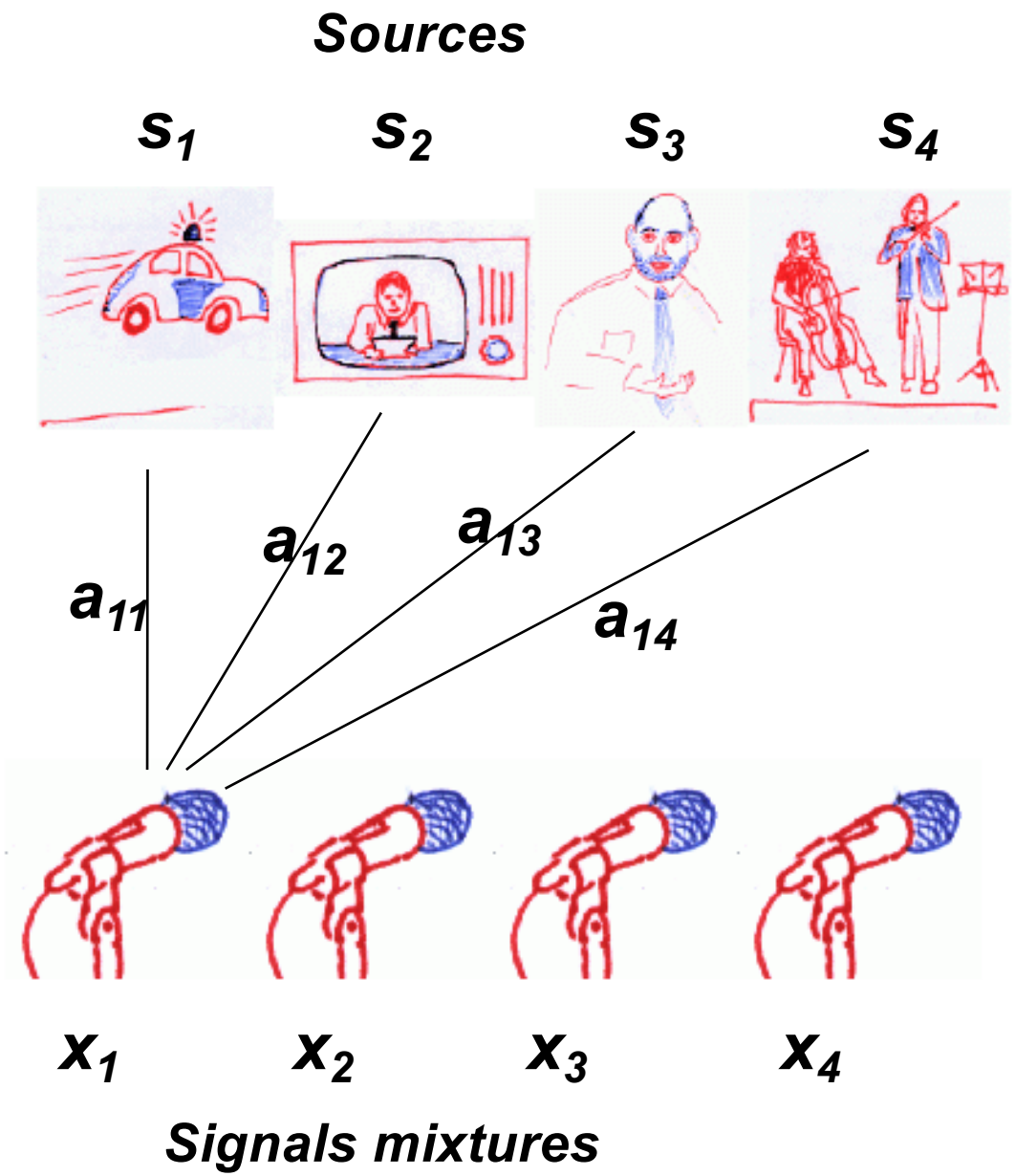


What is ICA?

"Independent component analysis (ICA) is a method for finding **underlying factors** or components from multivariate (multi-dimensional) statistical data. What distinguishes ICA from other methods is that it looks for components that are **both *statistically independent*, and *nonGaussian***."

A.Hyvarinen, A.Karhunen, E.Oja
'Independent Component Analysis'

The ICA model



$$x_i(t) = a_{i1} * s_1(t) + a_{i2} * s_2(t) + a_{i3} * s_3(t) + a_{i4} * s_4(t)$$

Here, i=1:4.

In vector-matrix notation, and dropping index t, this is

$$\boldsymbol{x = A * s}$$

Matrix notation of the mixtures

ICA Definition

- ICA is a statistical method, the goal of which is to **decompose given multivariate data into a linear sum of statistically independent components**
- For example, given two-dimensional vector , $\boldsymbol{x} = [\, x_1\ x_2 \,]^T$, ICA aims at finding the following decomposition

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} s_1 + \begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix} s_2$$

$$\boldsymbol{x} = \boldsymbol{a}_1 s_1 + \boldsymbol{a}_2 s_2$$

where $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ are basis vectors and s_1, s_2 are basis coefficients

Constraint: Basis coefficients s_1 and s_2 are statistically independent

ICA Definition

- Mixed Signals in Matrix Notation

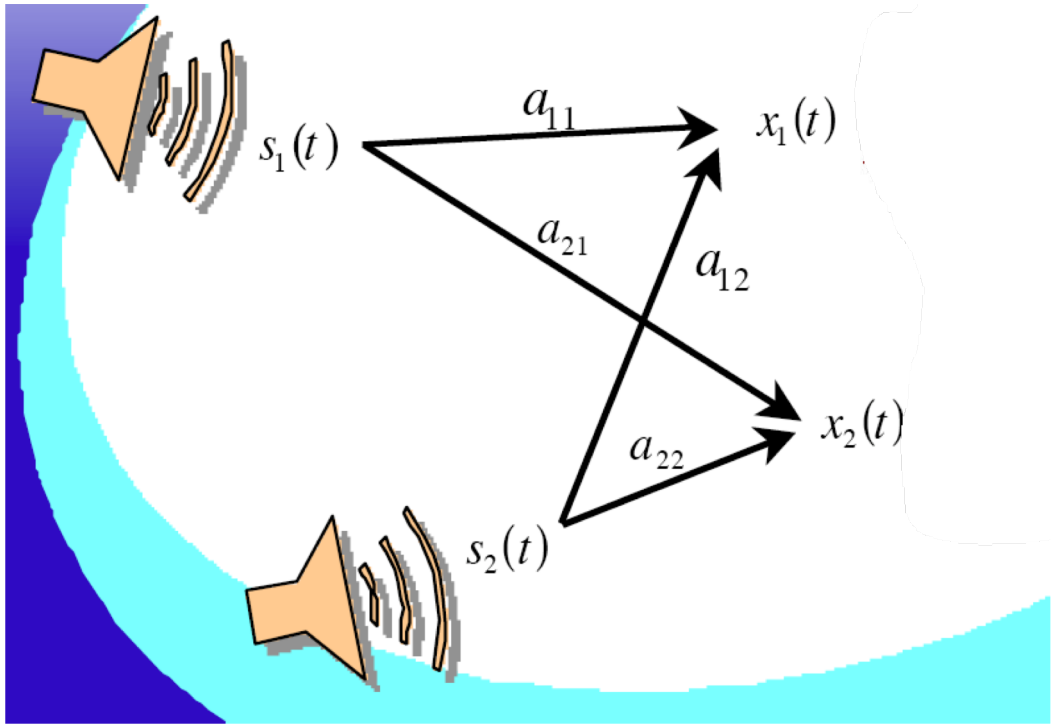
$$\mathbf{x} = \mathbf{A} \times \mathbf{s} = \sum_{i=1}^{n} \mathbf{a_i} \bullet s_i$$

- ***Variable Definitions***

$\mathbf{x_j} \sim$ Multiplexed Signal

$\mathbf{A} \sim$ Multiplexing Matrix

$\mathbf{s}_i \sim \mathrm{i}^{\mathrm{th}}$ Independent Signal



ICA Solution

- Signal Separation

Unximing matrix

$$\hat{\mathbf{s}} = \mathbf{W} \times \mathbf{x} = \mathbf{W} \times (\mathbf{A} \times \mathbf{s})$$

$$= (\mathbf{W} \times \mathbf{A}) \times \mathbf{s} = \mathbf{I} \times \mathbf{s} = \mathbf{s}$$

  - Find $\mathbf{W}$ using the ICA Algorithm

Goal will be to find a matrix W such that the entries of $\hat{S} = WX$ *are* ***as independent as possible.***

ICA Separation Technique

- **Central Limit Theorem**
  - If two random (non-Gaussian) signals are added, the resulting signal will be more Gaussian than the original two random signals

- **ICA Separation Concept**
  - Central Limit Theorem (in Reverse)
  - Maximizing Non-Gaussianity
    - Results in separating the two signals

ICA Assumption #1: Independence

- **Probability Density Definition**

$$p(y_1, y_2) = p_1(y_1)\,p_2(y_2)$$

$$p_1, p_2 \text{ are pdf's}$$

- **Expected Value Definition**

$$E\{h_1(y_1) * h_2(y_2)\} = E\{h_1(y_1)\} * E\{h_2(y_2)\}$$

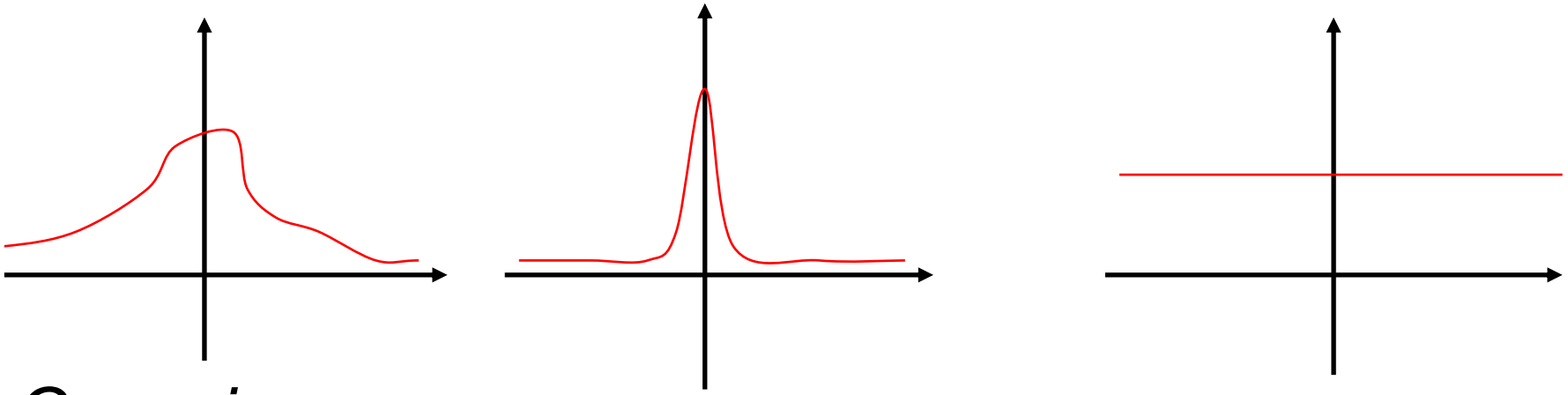
$$h_1, h_2 \text{ are functions}$$

What is independent component?

- If one variable can not be estimated from other variables, it is independent.
- By **Central Limit Theorem**, a sum of two independent random variables is more gaussian than original variables ➔ distribution of independent components are nongaussian
- To estimate ICs, **y** should have nongaussian distribution, i.e. we should **maximize nonguassianity.**

What is nongaussianity?

- Supergaussian
- Subgaussian
- Low entropy



ICA Assumption #2: Non-Gaussian

- **Property of Gaussian signals**
  - Addition of two independent Gaussian random variables is another single Gaussian random variable.
  - Information Lost!

- **Measuring nongaussianity by Kurtosis Function**
  - Kurtosis : 4th order cumulant of random variable

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2$$

  - Special case Kurt(y) = 0

How to measure nongaussianity?

- Measuring nongaussianity by Kurtosis
  - Kurtosis : 4th order cumulant of random variable

$$kurt(z) = E\{z^4\} - 3(E\{z^2\})^2$$

  - If kurt(z) is zero, gaussian
  - If kurt(z) is positive, supergaussian
  - If kurt(z) is negative, subgaussian

- Maximization of |kurt(z)| by gradient method

$$\frac{\partial |kurt(\mathbf{w}^T\mathbf{x})|}{\partial \mathbf{w}} = 4sign(kurt(\mathbf{w}^T\mathbf{x}))[E\{\mathbf{x}(\mathbf{w}^T\mathbf{x})^3\} - 3\mathbf{w}\,\|\mathbf{w}\|^2]$$

Simply change The norm of w

$$\Delta\mathbf{w} \propto sign(kurt(\mathbf{w}^T\mathbf{x}))E\{\mathbf{x}(\mathbf{w}^T\mathbf{x})^3\}$$

$$\mathbf{w} \leftarrow \mathbf{w}/\|\mathbf{w}\|$$

$$\mathbf{w} \leftarrow E\{\mathbf{x}(\mathbf{w}^T\mathbf{x})^3\} - 3\mathbf{w}$$

Fast-fixed point algorithm

Measures of non-Gaussianity

- **Kurtotis** : gauss=0 (sensitive to outliers)
- **Entropy** : gauss=largest
- **Negentropy** : gauss = 0
  - (difficult to estimate)

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2$$

$$H(y) = -\int f(y) \log f(y) dy$$

- Approximations

$$J(y) = H(y_{gauss}) - H(y)$$

$$J(y) = \frac{1}{12} E\{y^2\}^2 + \frac{1}{48} kurt(y)^2$$

$$J(y) \approx \left[E\{G(y)\} - E\{G(v)\}\right]^2$$

- where v is a standard gaussian random variable and :

$$G(y) = \frac{1}{a} \log \cosh(a.y)$$

$$G(y) = -\exp(-a.u^2 / 2)$$

Limitation #1: Scaling

- ICA maximizes independence between signals.

$$E\{\lambda_1 h_1(\hat{X}_1) * \lambda_2 h_2(\hat{X}_2)\} = \lambda_1\lambda_2 E\{h_1(\hat{X}_1) * h_2(\hat{X}_2)\}$$

$$= \lambda_1 E\{h_1(\hat{X}_1)\} * \lambda_2 E\{h_2(\hat{X}_2)\}$$

Assuming the Separated Signals are Independent

Scaling Factor for Signal #1

Scaling Factor for Signal #2

Limitation #2: Signal Permutations

- The mixing matrix and independent components are unknown.

$$\mathbf{x} = \mathbf{A} * \mathbf{s} = \mathbf{A} * \mathbf{I} * \mathbf{s} = \mathbf{A} * (\mathbf{P}^{-1} * \mathbf{P}) * \mathbf{s}$$

$$= (\mathbf{A} * \mathbf{P}^{-1}) * (\mathbf{P} * \mathbf{s}) = \mathbf{A}' * \mathbf{s}'$$

$$\hat{\mathbf{x}} = \mathbf{W}' * \mathbf{A}' * \mathbf{s}' = (\mathbf{P} * \mathbf{W}) * (\mathbf{A} * \mathbf{P}^{-1}) * \mathbf{s}'$$

$$= (\mathbf{P} * (\mathbf{W} * \mathbf{A}) * \mathbf{P}^{-1}) * \mathbf{s}' = (\mathbf{P} * (\mathbf{I} * \mathbf{P}^{-1})) * \mathbf{s}'$$

$$= (\mathbf{P} * \mathbf{P}^{-1}) * \mathbf{s}' = \mathbf{I} * \mathbf{s}' = \mathbf{s}' = \mathbf{P} * \mathbf{s}$$

Limitation #3: No. of Sensors

- Sensor Requirement
  - The number of separated signals cannot be larger than the number of inputs.
  - Current research is being done to reduce this constraint.

Application domains of ICA

- Blind source separation (BSS)
- Image denoising
- Medical signal processing – fMRI, ECG, EEG
- Modelling of the hippocampus and visual cortex
- Feature extraction, face recognition
- Compression, redundancy reduction
- Watermarking
- Clustering
- Time series analysis (stock market, microarray data)
- Topic extraction
- Econometrics: Finding hidden factors in financial data

ICA in Python

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
from sklearn.decomposition import FastICA, PCA

# Generate sample data
np.random.seed(0)
n_samples = 2000
time = np.linspace(0, 8, n_samples)

s1 = np.sin(2 * time)  # Signal 1 : sinusoidal signal
s2 = np.sign(np.sin(3 * time))  # Signal 2 : square signal
s3 = signal.sawtooth(2 * np.pi * time)  # Signal 3: saw tooth signal

S = np.c_[s1, s2, s3]
S += 0.2 * np.random.normal(size=S.shape)  # Add noise
```

ICA in Python

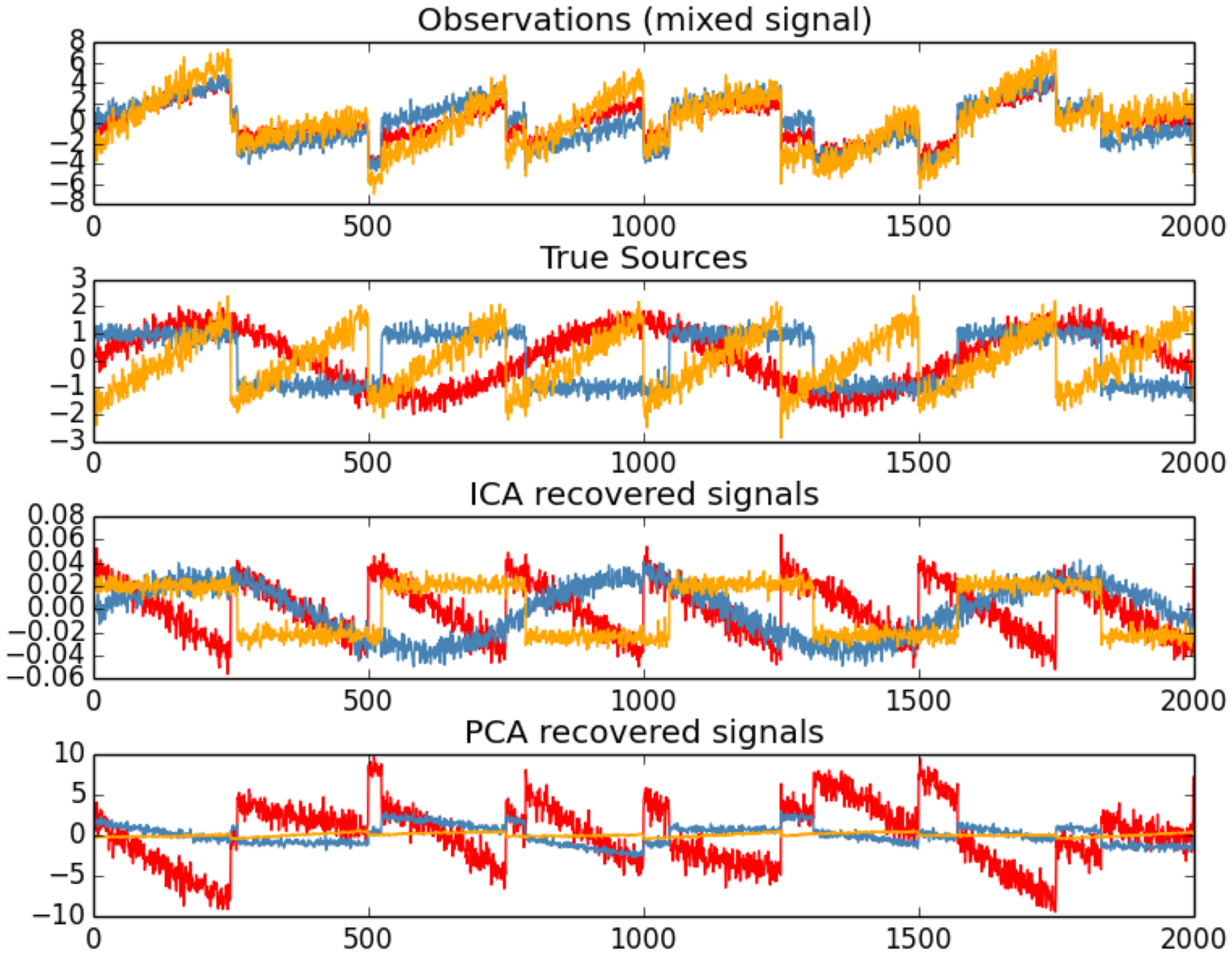
```
S /= S.std(axis=0)  # Standardize data
# Mix data
A = np.array([[1, 1, 1], [0.5, 2, 1.0], [1.5, 1.0, 2.0]])  # Mixing matrix
X = np.dot(S, A.T)  # Generate observations


# Compute ICA
ica = FastICA(n_components=3)
S_ = ica.fit_transform(X)  # Reconstruct signals
A_ =ica.mixing _  # Get estimated mixing matrix

# We can `prove` that the ICA model applies by reverting the unmixing.
assert np.allclose(X, np.dot(S_, A_.T) + ica.mean_)
```

ICA in Python

```
# For comparison, compute PCA
pca = PCA(n_components=3)
H = pca.fit_transform(X)  # Reconstruct signals based on orthogonal components
# Plot results
plt.figure()
models = [X, S, S_, H]
names = ['Observations (mixed signal)',  'True Sources', 'ICA recovered signals',
         'PCA recovered signals']
colors = ['red', 'steelblue', 'orange']
for ii, (model, name) in enumerate(zip(models, names), 1):
    plt.subplot(4, 1, ii)
    plt.title(name)
    for sig, color in zip(model.T, colors):
        plt.plot(sig, color=color)
plt.subplots_adjust(0.09, 0.04, 0.94, 0.94, 0.26, 0.46)
plt.show()
```

UNIVERSITAT DE BARCELONA

ICA in Python (Example)



Links

TO READ:

Naik, G.R., "Introduction: Independent Component Analysis"

Read Section 1.

The remaining is optional

TO READ: (optional)

Tharwat, A. (2020), "Independent component analysis: An introduction", Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2018.08.006

IT DETAILS THE SAME BUT EXPLAINS MORE EXPLICITLY THE MATHEMATICAL FOUNDATIONS OF ICA

Summary

- PCA : Proper to dimension reduction
- ICA : Proper to blind source separation or classification using Independent components when class id of training data is not available

Week 4

Course. Introduction to Machine Learning

Theory 4. Factor Analysis

Dr. Maria Salamó Llorente

maria.salamo@ub.edu

Dept. Mathematics and Informatics,

Faculty of Mathematics and Informatics,

University of Barcelona (UB)