GE461
Introduction to Data Science

Data Pre-processing

Shervin R. Arashloo

s.rahimzadeh@cs.bilkent.edu.tr

Spring 2022

Summary:

- Normalisation
- Data Cleaning: Noise Removal (Filtering)
- Data Cleaning: Anomaly Detection
- Data Compression: Karhunen-Loeve Transform
- Data Cleaning: Noise Removal (ICA)

Why Preprocess Data?

Data can be high in volume and come from multitude of sources and have variety of attributes

- Real-world data may be noisy, incomplete, inconsistent, corrupted, have missing values or attributes, outliers or conflicting values, etc.

- Analytical models fed with poor quality data can lead to poor or misleading predictions
- Quality decisions must be based on quality data: no quality data, no quality results!

- Data preparation stage tries to resolve such kinds of data issues to ensure the dataset is acceptable and of sufficient quality

- Data extraction, cleaning, and transformation comprises the majority of the work of building a data set

- Data preprocessing includes cleaning, instance selection, normalisation, transformation, feature extraction and selection, etc.

- The product of data preprocessing is the final training set

What Are the Benefits?

- Good data preparation is crucial to producing valid and reliable models that have <u>high accuracy and efficiency</u>

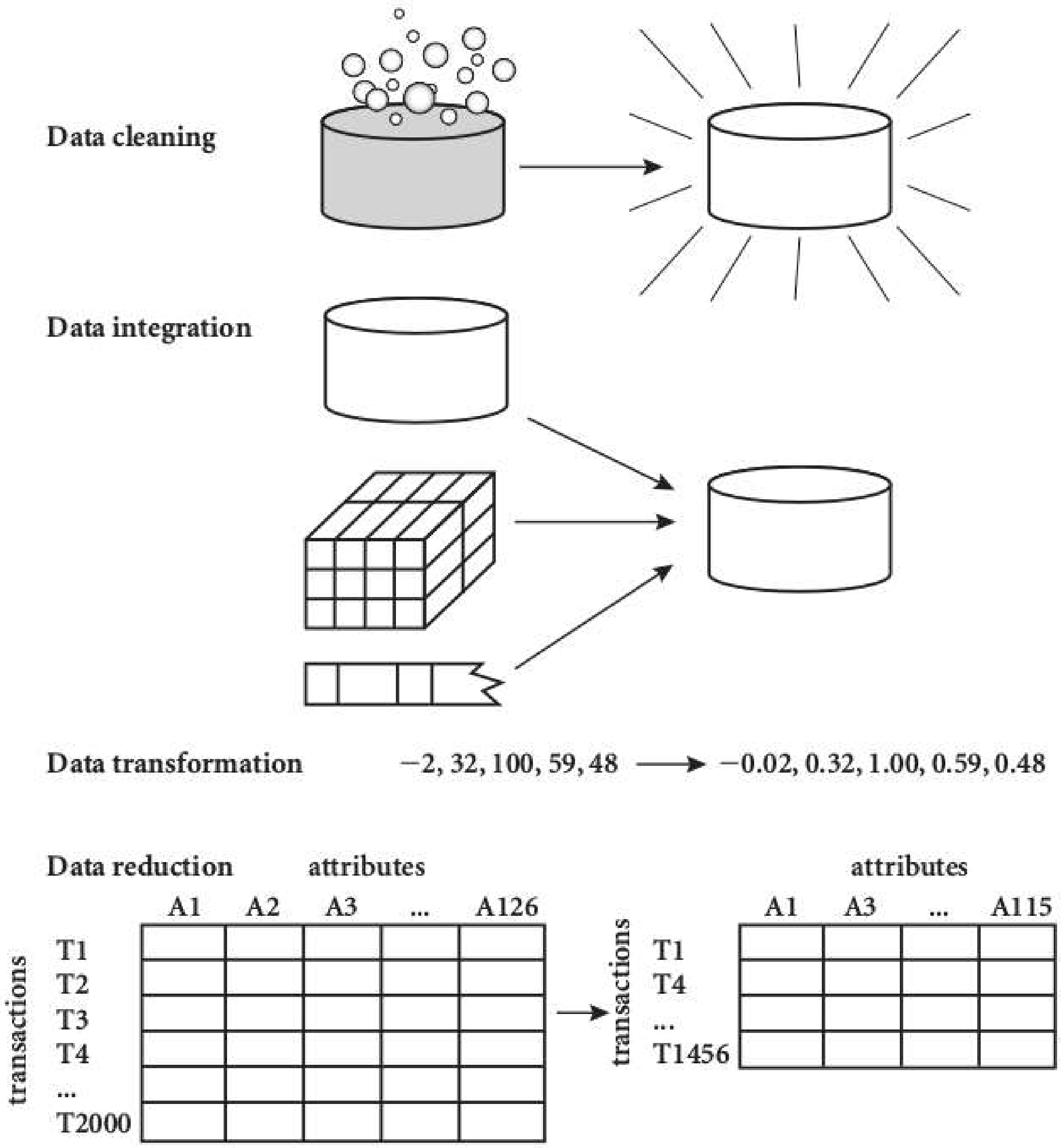
- It is essential to spot data issues early to <u>avoid getting misleading predictions</u>

- High quality data leads to <u>more useful insights</u> which enhance organisational decision making and improve overall <u>operational efficiency</u>

- Data preparation conducted cautiously and with analytical mindset can <u>save lots of time and effort</u>, and hence the costs incurred

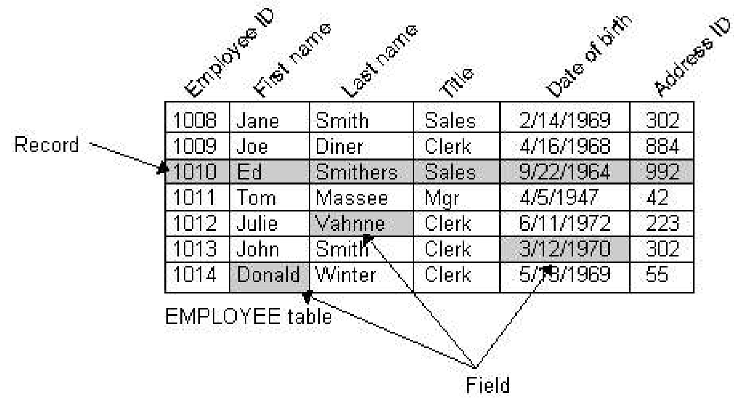
Data Preparation Activities

Data Preparation Activities	What to do?	How to do?
Data Cleaning	Dealing with Missing Values/Features	• Ignore respective records having missing values or features • Substitute with dummy value, mean, mode, regressed values or values predicted by an algorithm
	Dealing with Duplicate values/ Redundant Data	• Deletion of duplicate or redundant records
	Dealing with Outliers and Noise	• Binning • Regression (smoothing or curve fitting) • Clustering (grouping values in cluster to identify and eliminate outliers)
	Dealing with Inconsistent/ Conflicting Data	• Use of domain expertise, business understanding, human discretion to correct the data
Data Integration (Integrate multiple sources)	Dealing with issues like Schema integration, entity identification and redundancy	• Joining data sets • Editing metadata to handle data inconsistencies like naming, type etc.
Data Transformation	• Generalization of data	• Concept hierarchy climbing to replace low level attributes with high level concepts or attributes (ex. 'Street' can be generalized to 'country')
	• Normalization/ Scaling of attribute values to a specified range	• Z-score method • Min-Max method • Decimal scaling
	• Aggregation	• Applying summary or aggregation operators to data (ex. Using daily sales to compute annual sales)
	• Feature Construction	• Add or replace with new features derived from existing ones
Data Reduction (Reducing data to make it easy to handle and produce similar analytical results)	• Dimensionality Reduction to eliminate insignificant features	• Feature Selection • Attribute Sampling • Heuristic Methods
	• Aggregation	• Use of aggregation techniques (as above)
	• Data Compression	• Reducing data size by using methods like wavelet transform, PCA etc.
	• Numerousity reduction to have smaller data representations	• Record Sampling, Clustering, Regression etc.
	• Generalization	• Concept hierarchy generation (as above)
Data Discretization (cont. features into discrete)	• Unsupervised (no label is used)	• Binning (equal-width and equal-depth)
	• Supervised (uses labels)	• Entropy-based
Feature Engineering	• Using or deriving the right features to improve accuracy of your analytical model	• Feature Selection • Validation & improvement of features • Brainstorming to create and test more features



Data cleaning
Data integration
Data transformation
−2, 32, 100, 59, 48
−0.02, 0.32, 1.00, 0.59, 0.48
Data reduction
attributes
A1
A2
A3
...
A126
transactions
T1
T2
T3
T4
...
T2000
attributes
A1
A3
...
A115
transactions
T1
T4
...
T1456

Types of Data Sets



Employee ID	First name	Last name	Title	Date of birth	Address ID
1008	Jane	Smith	Sales	2/14/1969	302
1009	Joe	Diner	Clerk	4/16/1968	884
1010	Ed	Smithers	Sales	9/22/1964	992
1011	Tom	Massee	Mgr	4/5/1947	42
1012	Julie	Vahnne	Clerk	6/11/1972	223
1013	John	Smith	Clerk	3/12/1970	302
1014	Donald	Winter	Clerk	5/18/1969	55

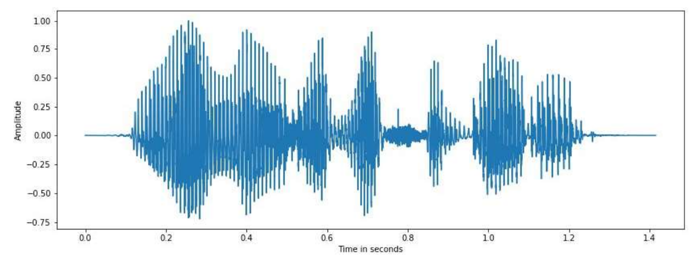
Record

- Relational records
- Data matrix, e.g. numerical matrix
- Document data: text documents
- Transaction data

Graph and network

- World Wide Web
- Social or information networks
- Molecular Structures

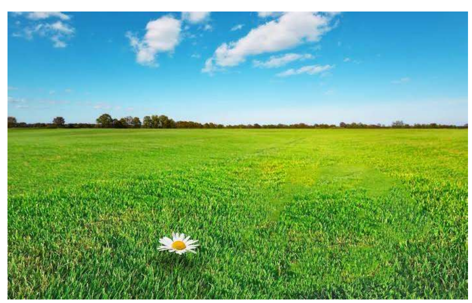
Ordered



- Video data: sequence of images
- Temporal data: time-series
- Sequential Data: transaction sequences
- Genetic sequence data

Spatial, image and multimedia

- Spatial data: maps
- Image/video data



Data Object

- Data sets are made up of **data objects,** representing an entity
- Examples:
  - sales database: customers, store items, sales
  - medical database: patients, treatments
  - university database: students, professors, courses
- Also called samples, examples, instances, data points, tuples, etc.

- Data objects are described by **attributes,** a.k.a. **features**

- Database rows -> data objects; columns ->attributes

Data Object

- Data sets are made up of **data objects,** representing an entity
- Examples:
  - sales database: customers, store items, sales
  - medical database: patients, treatments
  - university database: students, professors, courses
- Also called samples, examples, instances, data points, tuples, etc.

- Data objects are described by **attributes,** a.k.a. **features**

- Database rows -> data objects; columns ->attributes

Features/Attributes

Feature (attribute, dimensions, variables): a data field, representing a characteristic or feature of a data object.

- e.g., customer _ID, name, address

Types:

- **Nominal**: categories, states, or “names of things”
  - Hair_color = {black, brown, blond, red, auburn, grey, white}
- **Binary**
  - Nominal attribute with only 2 states (0 and 1)
  - e.g. gender
  - convention: assign 1 to the more important state
- **Ordinal**
  - Values have a meaningful order (ranking) but magnitude between successive values is not known
  - e.g. Size = {small, medium, large}, grades, army rankings
- **Numeric**: quantitative
  - Interval-scaled
  - Ratio-scaled

Normalisation

- Distance measures like the Euclidean distance are very often used to measure similarity between features/attributes
- Each single attribute may be equally important but such geometric measures implicitly assign more weighting to features with large ranges than those with small ranges
- Normalisation is meant to remove such undesired effects

- **Linear scaling to unit range**

  yields a normalised value in the range [0 1]

  l: lower bound and *u*: upper bound

$$\tilde{x} = \frac{x - l}{u - l}$$

- **Linear scaling to unit variance**

  yields a zero mean and unit variance feature

  mu: sample mean and *sigma:* sample standard deviation of the feature

$$\tilde{x} = \frac{x - \mu}{\sigma}$$

- Assuming that each feature is normally distributed, the probability of normalised feature being in the [-1,1] range is 68%
- An additional shift and rescaling as

$$\tilde{x} = \frac{(x - \mu)/3\sigma + 1}{2}$$

  guarantees 99% of the normalised features are in the [0,1] range

- **Transformation to a Uniform random variable**

  Given a random variable x with cumulative distribution function C(x), the random variable resulting from the transformation x' = C(x) is uniformly distributed in the [0,1] range (can be simply shown)

- **Rank normalisation**

  Given a sample for a feature component for all feature items as x1,..., xn, find the order statistics and then replace each feature value by its normalised rank:

$$\tilde{x}_i = \frac{\underset{x_1,\ldots,x_n}{\operatorname{rank}}(x_i) - 1}{n - 1}$$

Normalisation after fitting distributions

- Sample values can be used to find estimates for the feature distributions to be used to find normalisation methods based particularly on those distributions

- After estimating the parameters of a distribution, the cut-off value that includes 99% of the feature values is found and the sample values are scaled and truncated so that each feature component have the same range

Normal distribution

- likelihood function for the parameters $$L(\mu, \sigma^2 \,|\, x_1, \ldots, x_n) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\sum_{i=1}^{n} (x_i - \mu)^2 / 2\sigma^2 \right)$$
- the parameter estimates are:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i \quad \text{and} \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})^2$$

- The cut-off value that includes 99% of the feature values may be found as

$$P(x \leqslant \delta_x) = P\left(\frac{x - \hat{\mu}}{\hat{\sigma}} \leqslant \frac{\delta_x - \hat{\mu}}{\hat{\sigma}} \right) = 0.99$$
$$\Rightarrow \delta_x = \hat{\mu} + 2.4\hat{\sigma}.$$

Some other possibilities for distribution fitting:

- Uniform
- Gamma
- Chi-squared
- Weibull
- Beta
- Cauchy
- etc.

Which normalisation scheme should one follow?

Data Cleaning- Noise Removal

- Noisy data is data that is corrupted, or distorted, or has a low Signal-to-Noise Ratio
- Improper procedures to subtract out the noise in data can lead to a false sense of accuracy or false conclusions

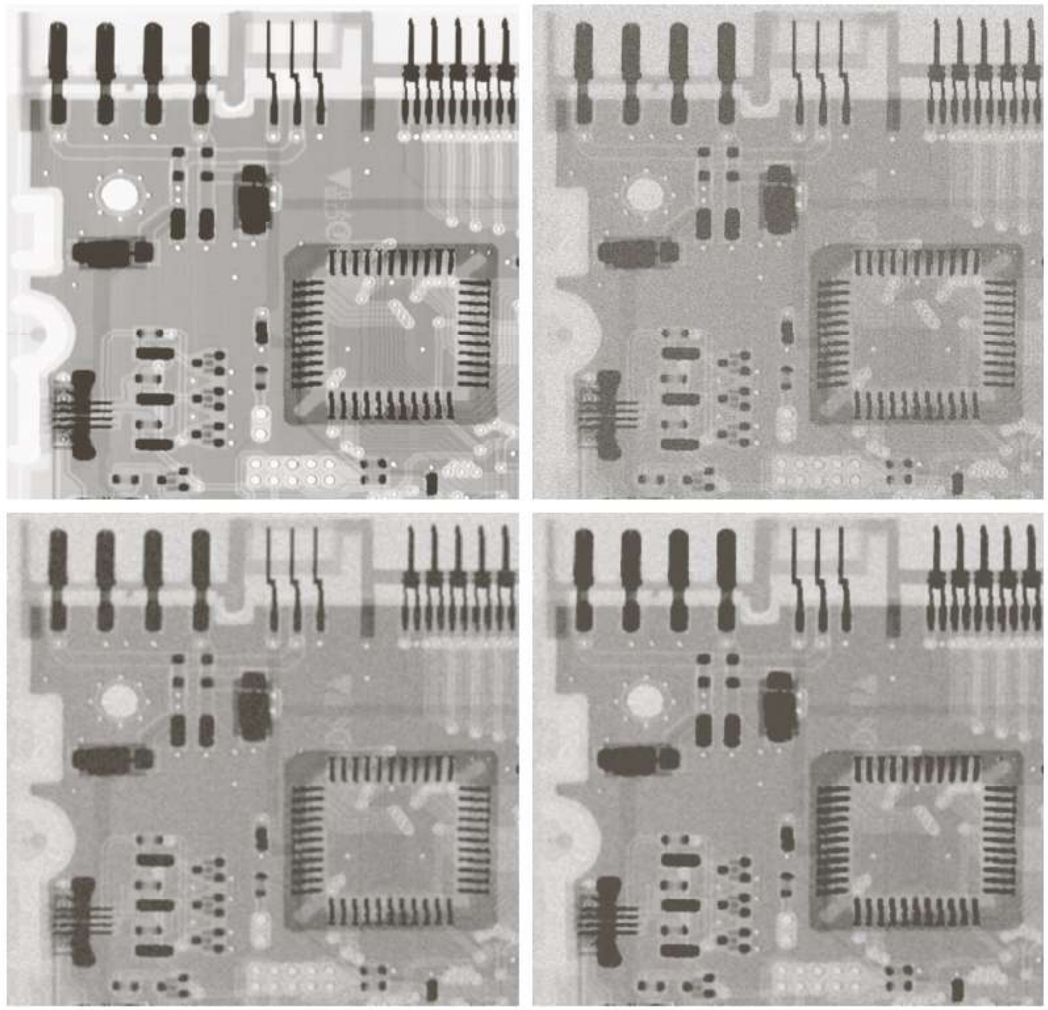
- In the presence of additive noise:

  Data = f(true signal) + noise, f(.) is a function

- Filtering may be used to remove/attenuate noise in the signal:
  - Temporal/spatial domain filtering
  - Frequency domain filtering

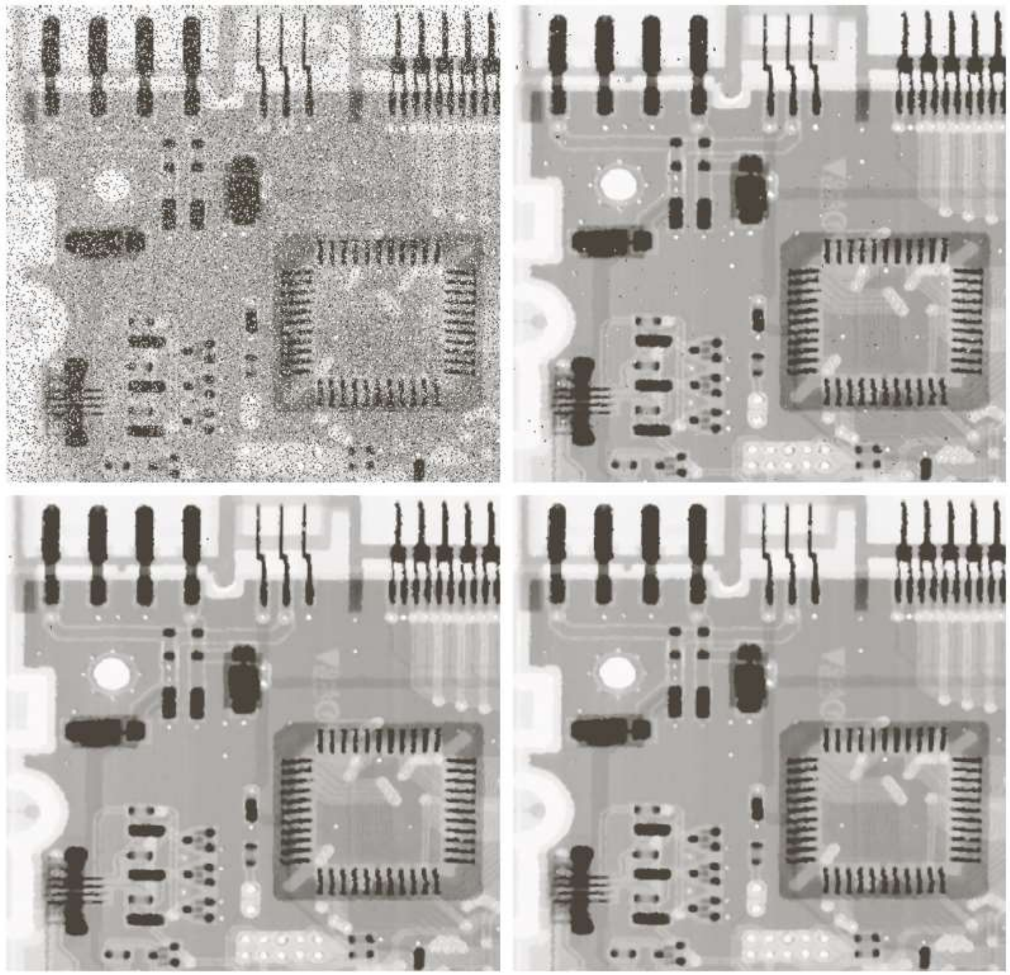
- **Spatial domain smoothing (lowpass) filters**
  - (arithmetic) Mean filtering: a data point is replaced by the average over the values in a pre-defined neighbourhood
  - Geometric mean filtering: a data point is replaced by the geometric mean over the values in a pre-defined neighbourhood

- **Spatial domain order-statistic filtering:**
  - Max filtering: a data point is replaced by the maximum over the values in a pre-defined neighbourhood
  - Min filtering: a data point is replaced by the minimum over the values in a pre-defined neighbourhood
  - Median filtering: a data point is replaced by the median over the values in a pre-defined neighbourhood



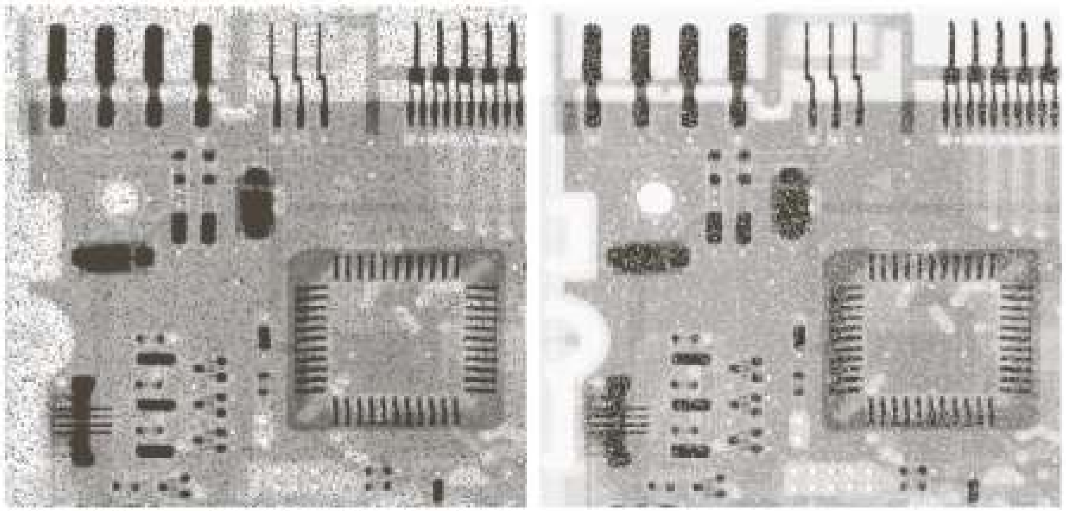
a b
c d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3×3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)



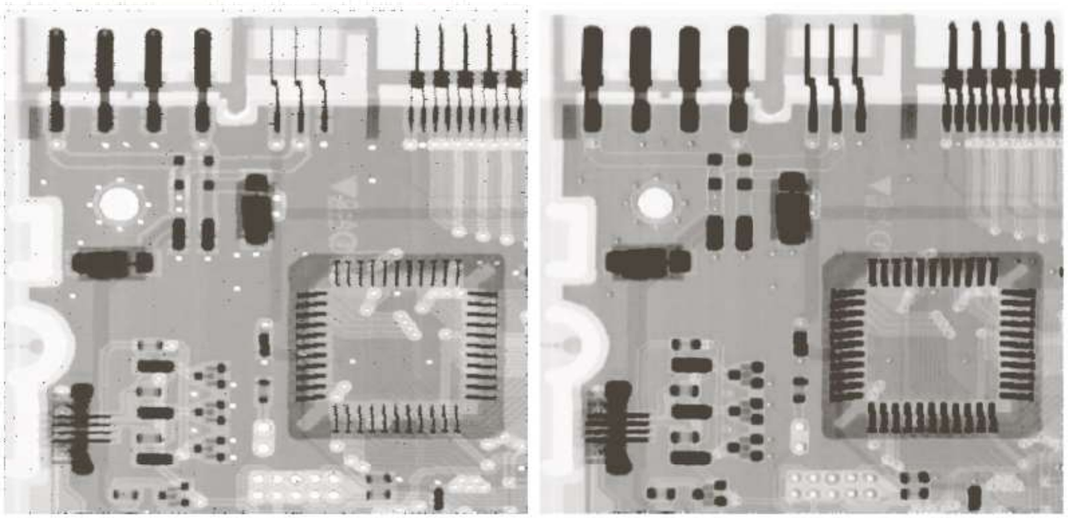
a b
c d

FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.



a b
c d

FIGURE 5.8
(a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt



a b

FIGURE 5.11
(a) Result of filtering Fig. 5.8(a) with a max filter of size 3×3. (b) Result of filtering 5.8(b) with a min filter of the same size.

Frequency domain filtering

The 2D Discrete Fourier Transform (DFT)

Defined for a sampled image *f(x, y)* of MxN pixels:

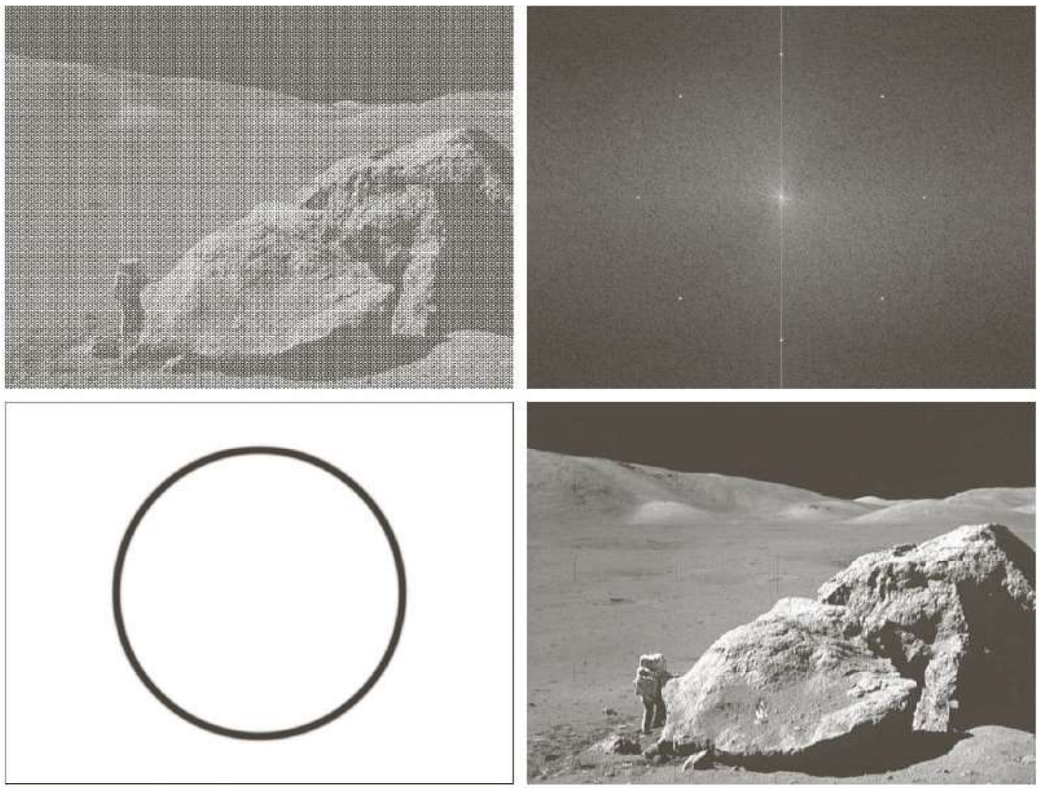
$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi(ux/M+vy/N)}$$

where x = 0, 1, 2…M-1, y = 0,1,2…N-1 and u = 0, 1, 2…M-1, v = 0, 1, 2…N-1.

How do you get back? Use the **Inverse transform**!

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi(ux/M+vy/N)}$$

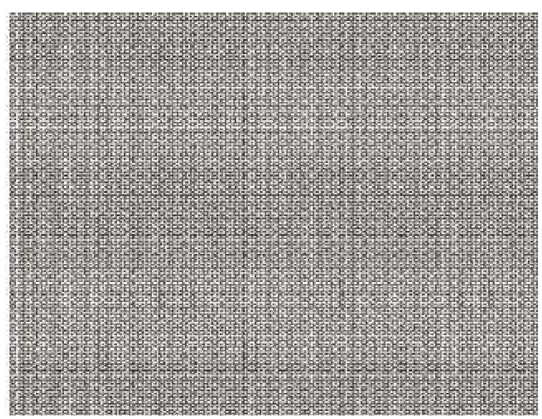
Fast changes in the original signal appear as high frequency components while slow changes in the signal appear as low frequency components



a b
c d

FIGURE 5.16
(a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

Noise pattern obtained by filtering →



Wiener Filtering

- The presumed signal model: g = f*h+n; "*" denotes convolution
- "g" is observed data, "f" is the original data, "h" is degradation function and "n" is the noise
- Minimise the mean squared error between the estimate $\hat{f}$ and the original signal f:

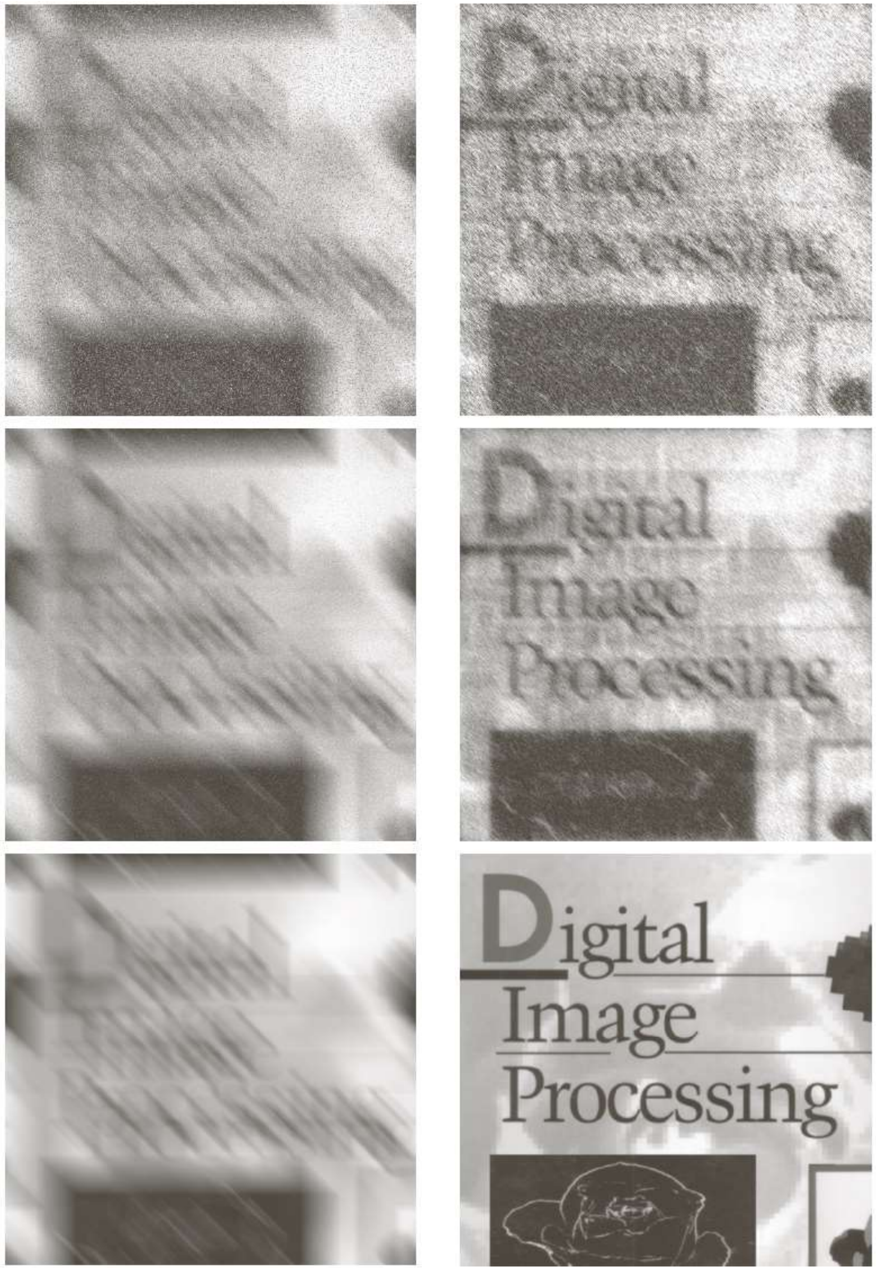
$$Minimise\ E(f-\hat{f})^2$$

  - where E(.) is the expected value of the argument

- The DFT of the estimated data can be shown to be equivalent to

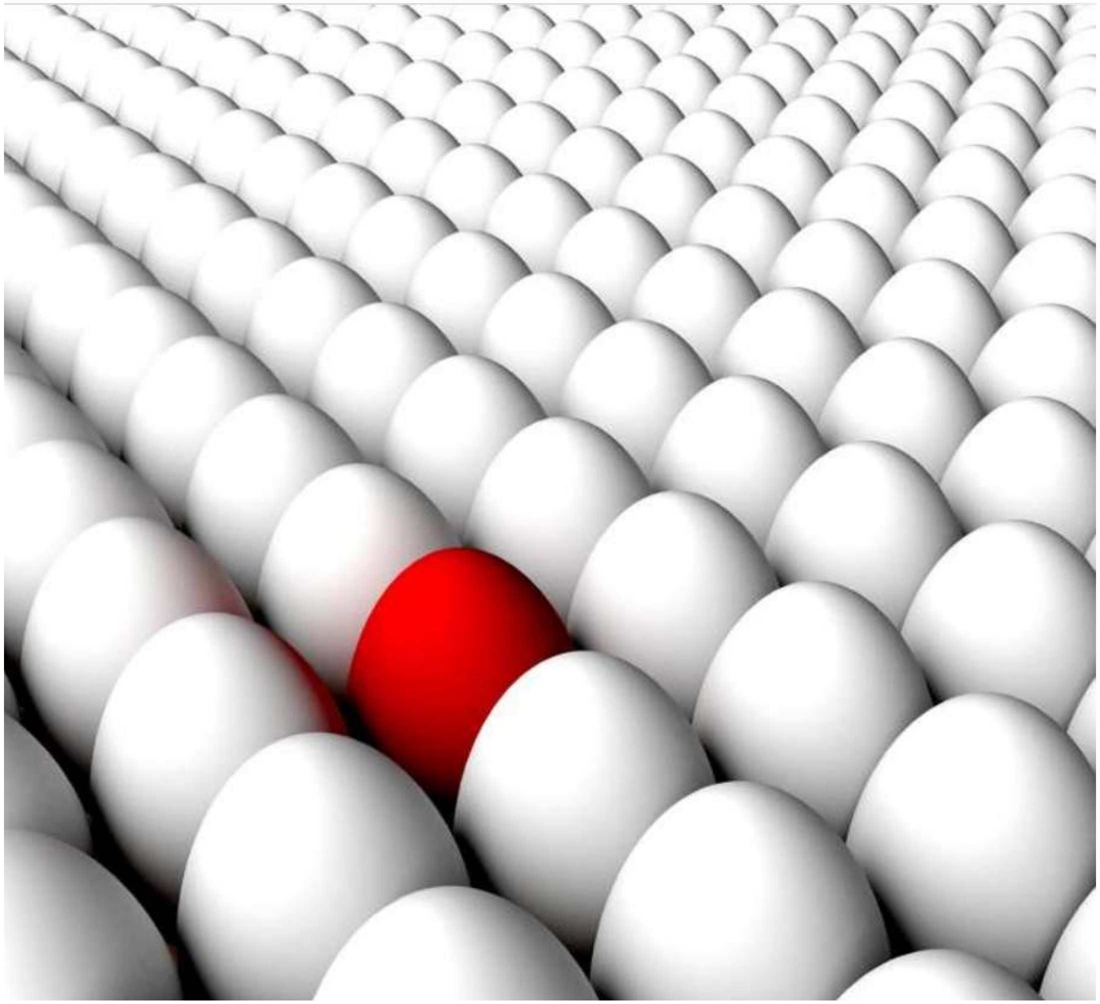
$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right]G(u,v)$$

- K is a constant reflecting the ratio of the noise and signal energies
- H(u,v) is approximated based on our prior assumption of the degradation function



Left column: image corrupted by additive noise and motion blur. Right column: result of Wiener filtering
* Noise levels decrease from top to bottom

Data Cleaning- Outlier Detection



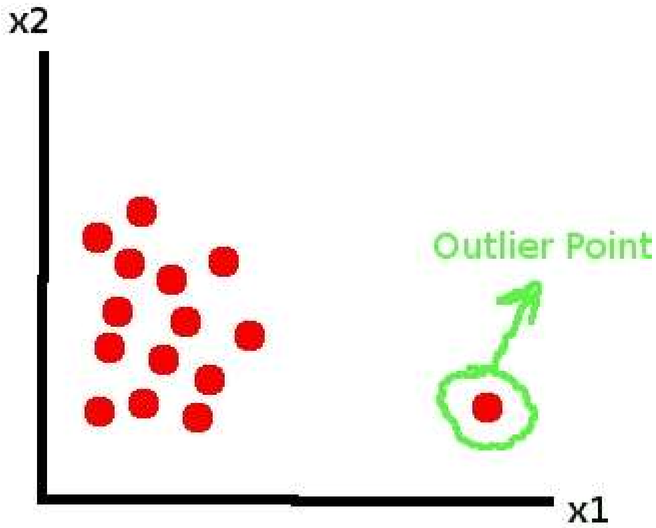
- What is an **outlier**?
  - "*An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism*" [1]
  - Closely related/synonymous terms: anomaly, novelty, surprise, etc.
- Outliers violate the mechanism that generates the normal data

- **Applications**:
  - Data cleaning - Medical analysis
  - Detecting measurement errors - Sports statistics
  - Public health - Fraud detection

[1] Hawkins D., "Identification of Outliers", Chapman and Hall, 1980

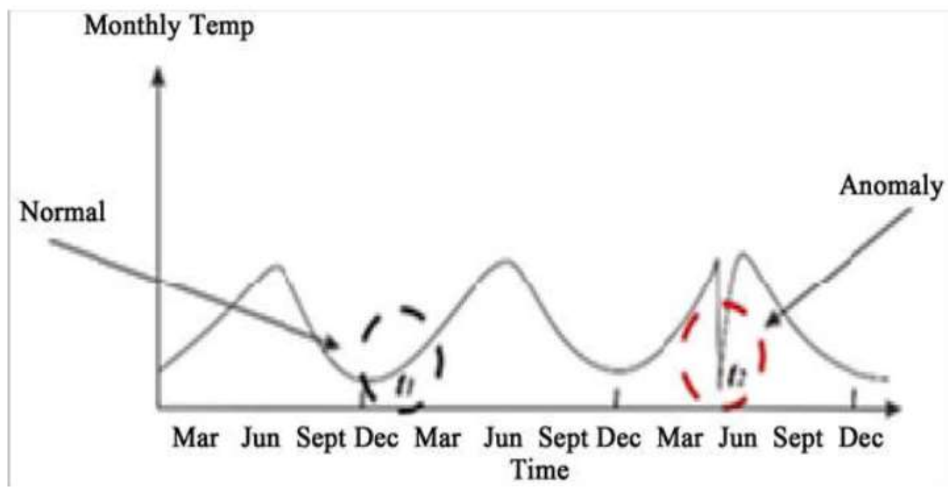
Types of Outliers/Anomalies

- **Point Anomaly**
  - An object that significantly deviates from the rest of the data set
  - Example: Intrusion in computer networks
- **Contextual Anomaly**
  - An object that deviates significantly based on a selected context
  - Example: temperature in a particular month



Point anomaly

Contextual anomaly



How to detect anomalies?

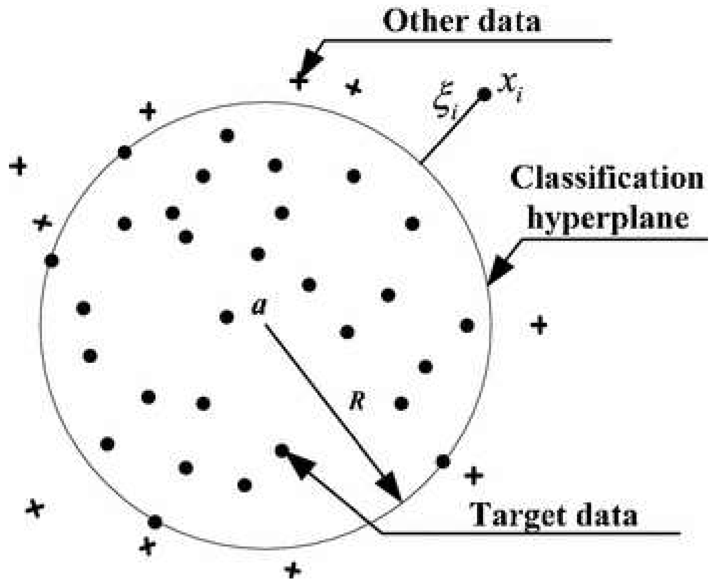
- Different algorithms may be categorised from the point of view of having access to different data types: normal only, outlier only, or both

- **One-Class Classification (OCC)**

  deals with the problem of identifying objects from the target/positive class, and distinguishing them from all other objects, typically known as outliers or anomalies

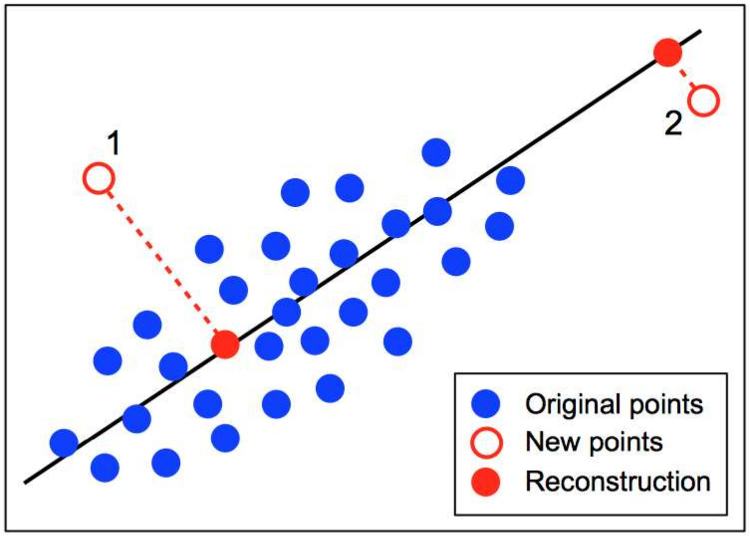
- Different OCC techniques:
  - Boundary methods
  - Reconstruction-based methods
  - Density-based methods

- In boundary-based approaches, the goal is to optimise a boundary encompassing the target set of objects
- Example technique: One-Class Support Vector Machine (OC-SVM)[2]

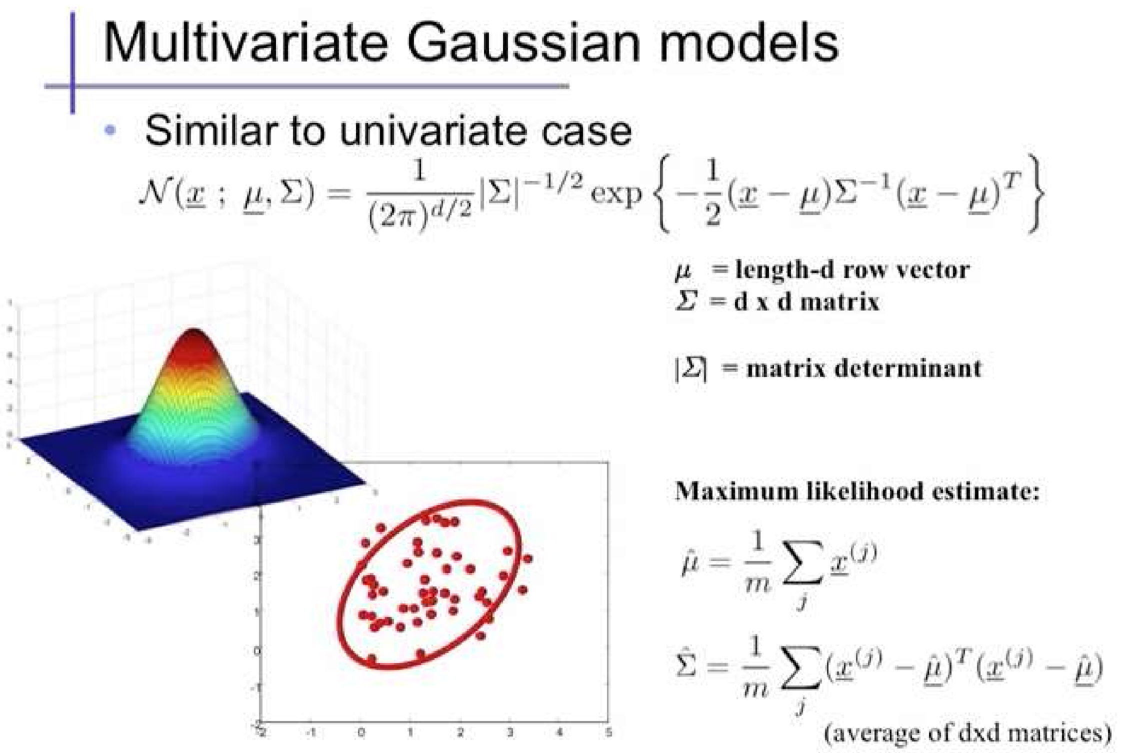


[2] Tax, D.M., Duin, R.P. Support Vector Data Description. Machine Learning 54, 45–66 (2004)

- In the reconstruction-based category, typically, a model is chosen and fit to the data which makes it viable to represent new objects in terms of their affinity to the generative model
- Detection is typically based on the reconstruction residual of an object using the presumed model
- Example technique: PCA (to be discussed)



- The density-based approaches try to estimate the probability density of the training data followed by setting a threshold on the estimated density
- Several different distributions have been assumed in practice including the Gaussian or a Poisson distribution



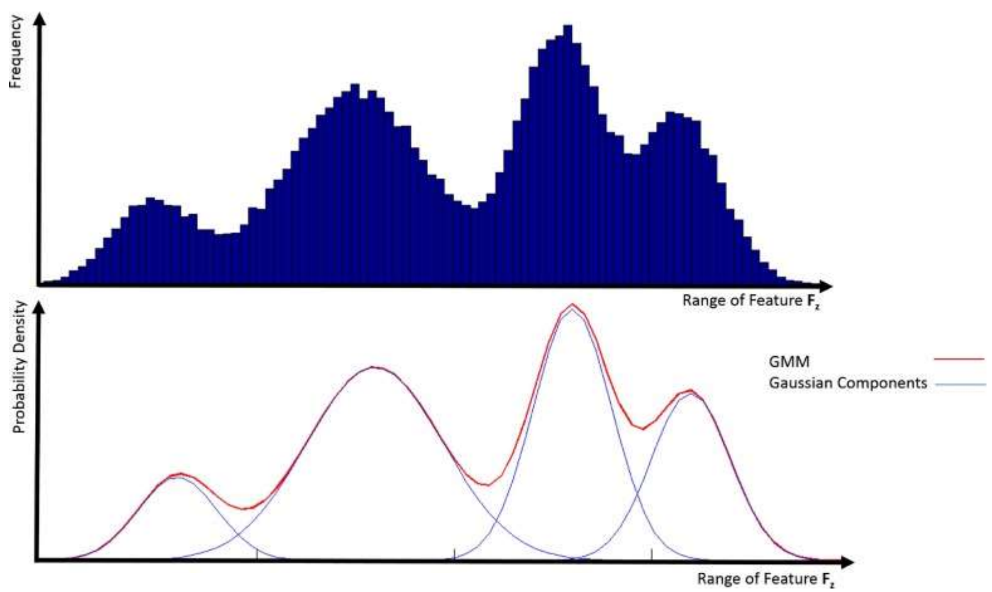
- The **Mahalanobis distance** is a measure of the distance between a point P and a distribution D
- It is a multi-dimensional generalization of the idea of measuring how many standard deviations away P is from the mean of D

- Squared Mahalanobis distance: $(x-\mu)^t \sum^{-1} (x-\mu)$
  - μ is the sample mean and $\sum^{-1}$ is the inverse covariance matrix of the data

- reduces to Euclidean distance if covariance matrix is identity
- The larger the Mahalanobis distance, the more likely to be an outlier!

Mixture Models

- Instead of a single parametric model, use multiple models to better capture the probability density function of the data
- Example: Gaussian Mixture Model (GMM) → Week 10
- For anomaly detection compute the minimum Mahalanobis distance of an observation to all mixture components



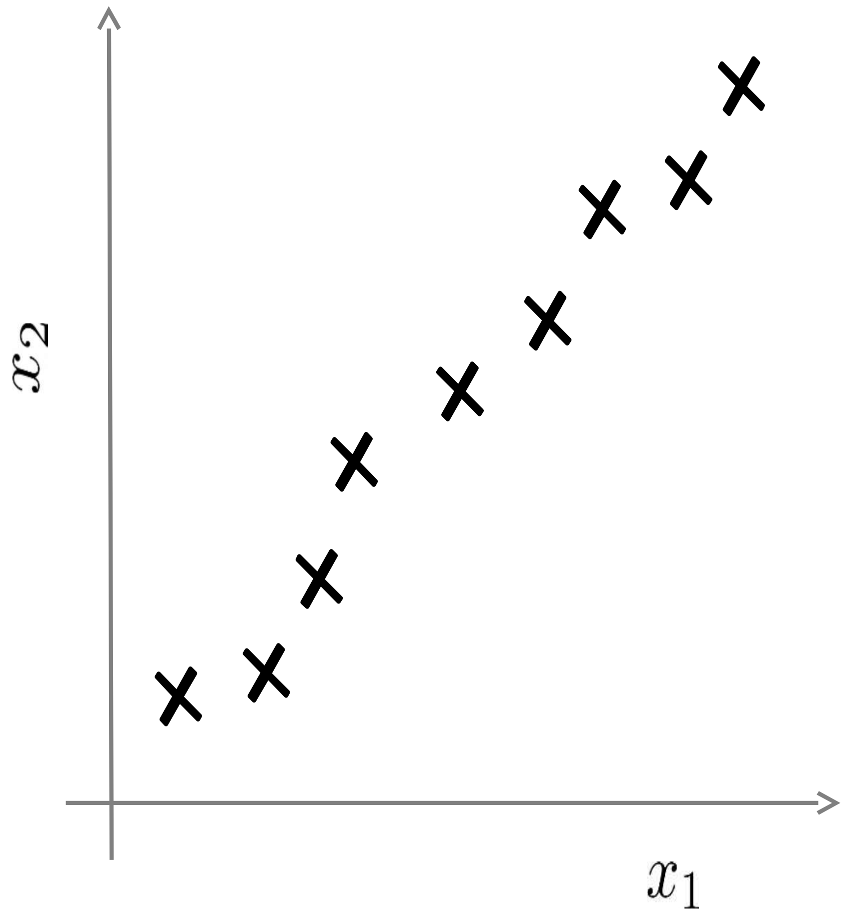
Karhunen-Loeve Transform
(Principal Component Analysis)

Motivation

- Can reduce the dimensionality: data **compression**
- A linear transformation of the data points
- Reduces the correlation between data points
- May reduce the noise
- May be used for data visualisation
- Unsupervised: no labels required

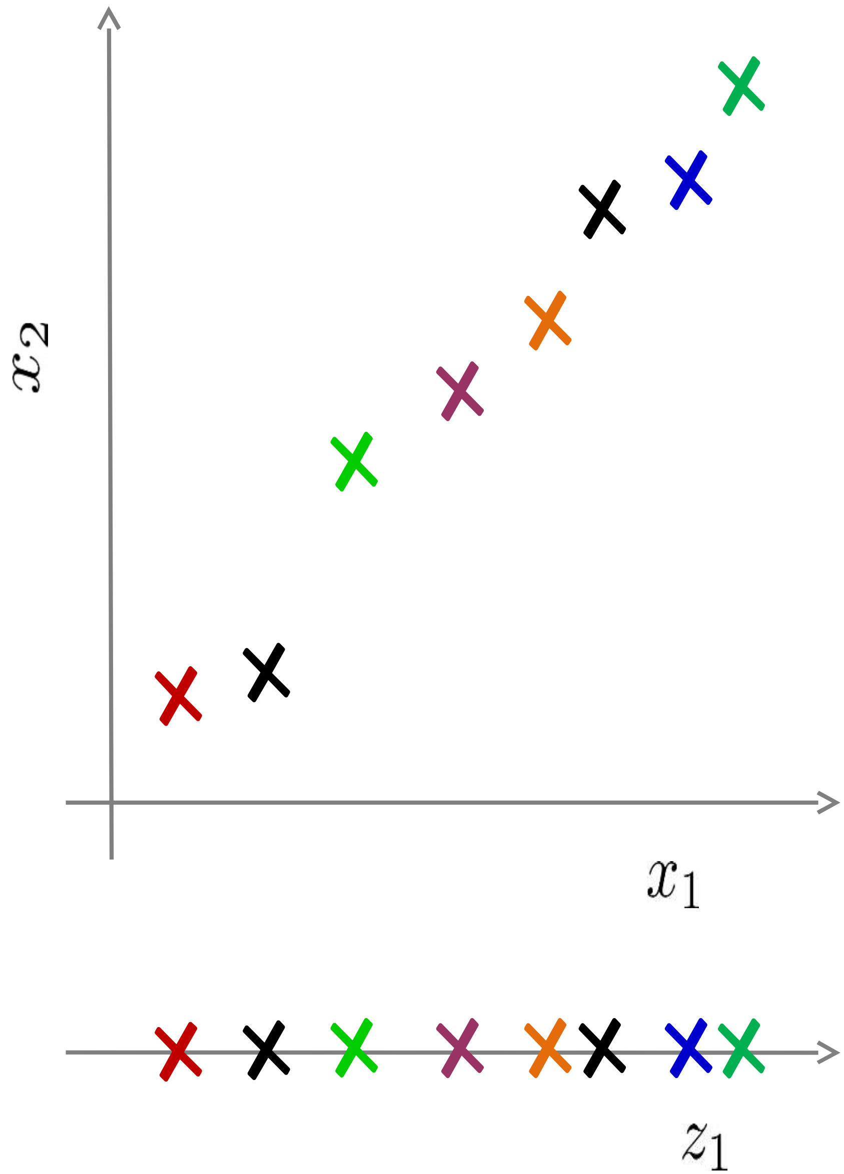
- Given data points in *d* dimensions
- Convert them to data points in *r*<*d* dimensions
- With minimal loss of information

Data Compression



Reduce data from 2D to 1D

Data Compression



Reduce data from 2D to 1D

$$x^{(1)} \rightarrow z^{(1)}$$

$$x^{(2)} \rightarrow z^{(2)}$$

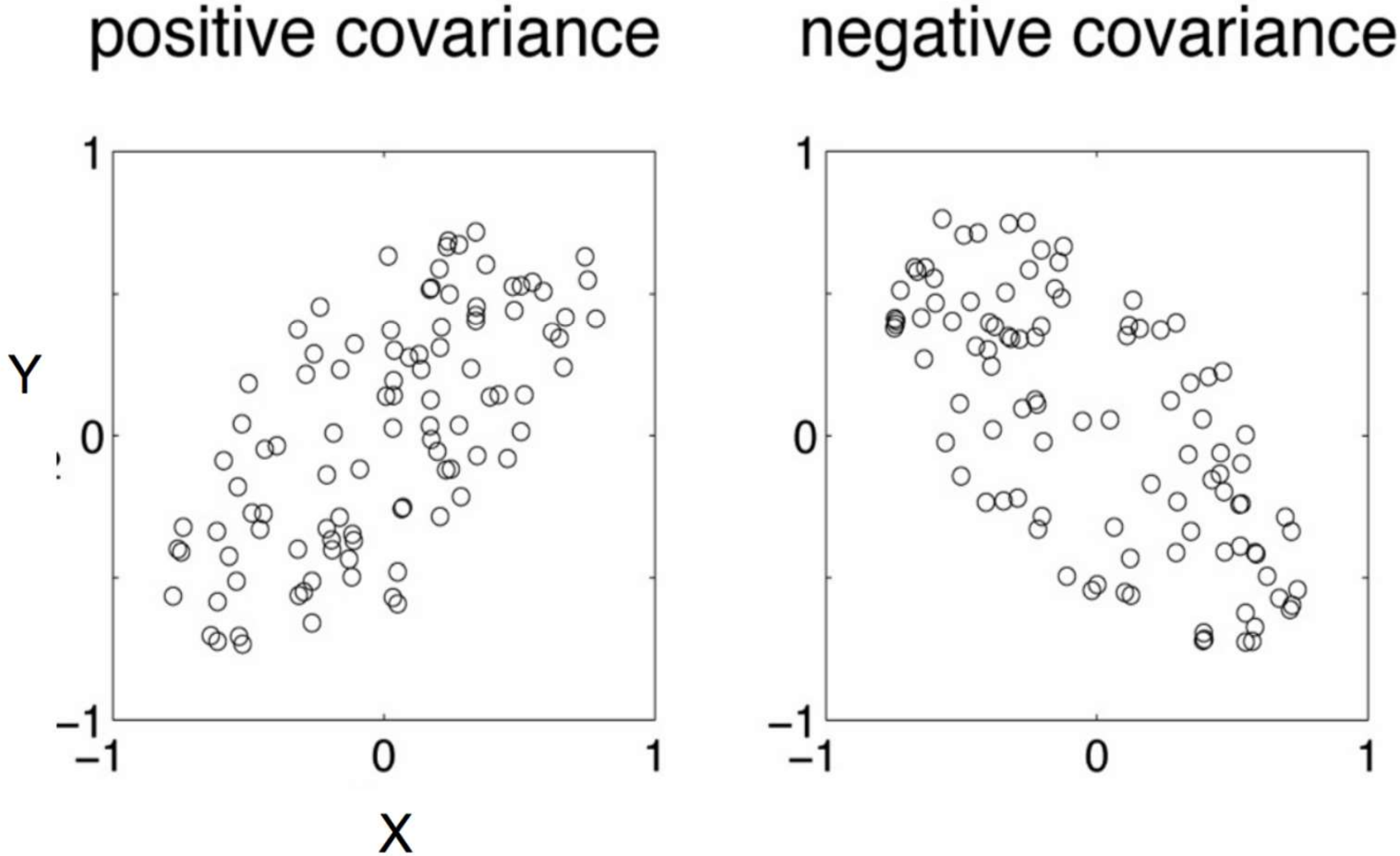
$$\vdots$$

$$x^{(m)} \rightarrow z^{(m)}$$

Covariance

- Variance and Covariance:
  - Measure of the "spread" of a set of points around their center of mass(mean)
- Variance:
  - Measure of the deviation from the mean for points in one dimension
- Covariance:
  - Measure of how much each of the dimensions vary from the mean with **respect to each other**

- **Covariance is measured between two dimensions**
- **Covariance sees if there is a relation between two dimensions**
- **Covariance between one dimension is the variance**



Positive: Both dimensions increase or decrease together

Negative: While one increase the other decrease

Eigenvector and Eigenvalue

$Ax = \lambda x$

A: Square Matirx
λ: Eigenvector or characteristic vector
X: Eigenvalue or characteristic value

- ***The zero vector can not be an eigenvector***
- ***The value zero can be eigenvalue***

The eigenvectors of the covariance matrix form the bases for the PCA transformation

The larger the eigenvalue of an eigenvector, the more important it is

Input: $\mathbf{x} \in \mathbb{R}^D$: $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$

Set of basis vectors: $\mathbf{u}_1, \ldots, \mathbf{u}_K$

Summarize a *D* dimensional vector *X* with *K* dimensional feature vector *h*(*x*)

$$h(\mathbf{x}) = \begin{bmatrix} \mathbf{u}_1 \cdot \mathbf{x} \\ \mathbf{u}_2 \cdot \mathbf{x} \\ \cdots \\ \mathbf{u}_K \cdot \mathbf{x} \end{bmatrix}$$

$$\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$$

ui's are the eigenvectors of the covariance matrix

New data representation *h(x)*

$$h(\mathbf{x}) = \mathbf{U}^T(\mathbf{x} - \mu_0)$$

Empirical mean of the data → $\mu_0 = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$

PCA steps

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

Application: Image compression

Original Image

- Divide the original 372x492 image into patches
- Each patch is an instance that contains 12x12 pixels on a grid
- View each as a 144-D vector

PCA compression: 144D $\Rightarrow$ 60D

PCA compression: 144D $\Rightarrow$ 16D

PCA compression: 144D $\Rightarrow$ 6D

PCA compression: 144D $\Rightarrow$ 3D

Independent Component Analysis

"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 *non-Gaussian*."

A.Hyvarinen, A.Karhunen, E.Oja

'Independent Component Analysis'

Independent Component Analysis (ICA) is the identification & separation of mixtures of sources with little prior information

Applications include:

Denoising

Blind source separation

Medical signal processing

Compression, redundancy reduction

Scientific Data Mining

etc.

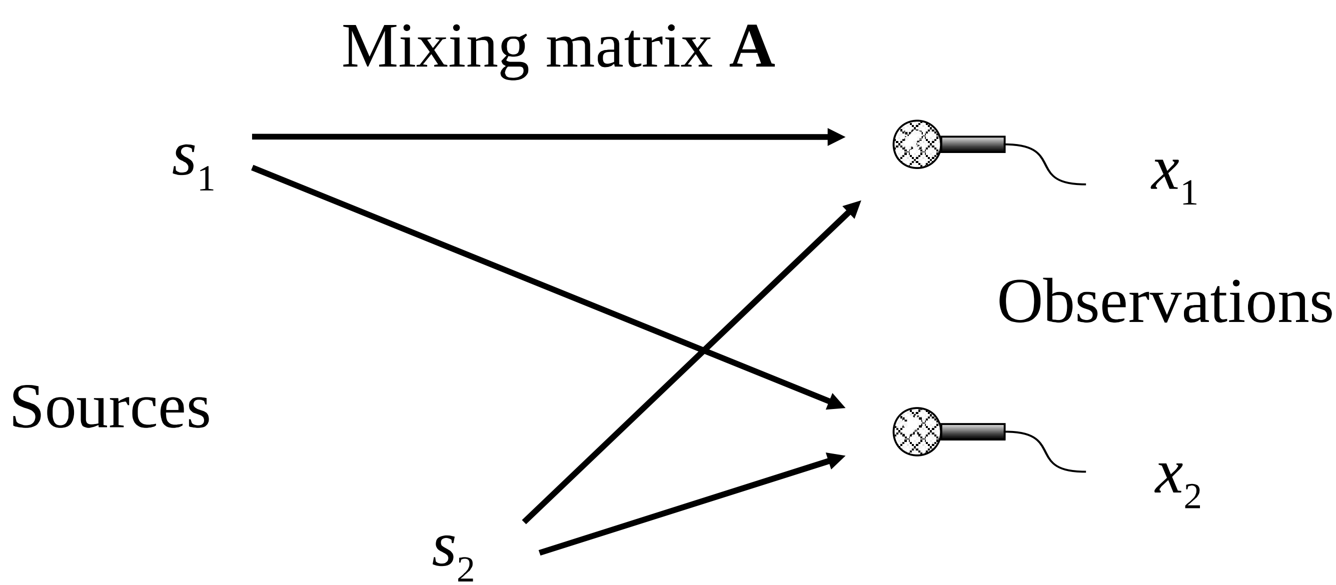
ICA seeks directions that are as **independent** from each other as possible

A set of observations of random variables $x_1(t), x_2(t)\ldots x_n(t)$, where t is the time or sample index

Assume that they are generated as a linear mixture of independent components: $\mathbf{x}=\boldsymbol{A}\boldsymbol{s}$, where $\boldsymbol{A}$ is some unknown matrix

Independent component analysis now consists of estimating both the matrix $\boldsymbol{A}$ and the $s_i(t)$, when we only observe the $x_i(t)$

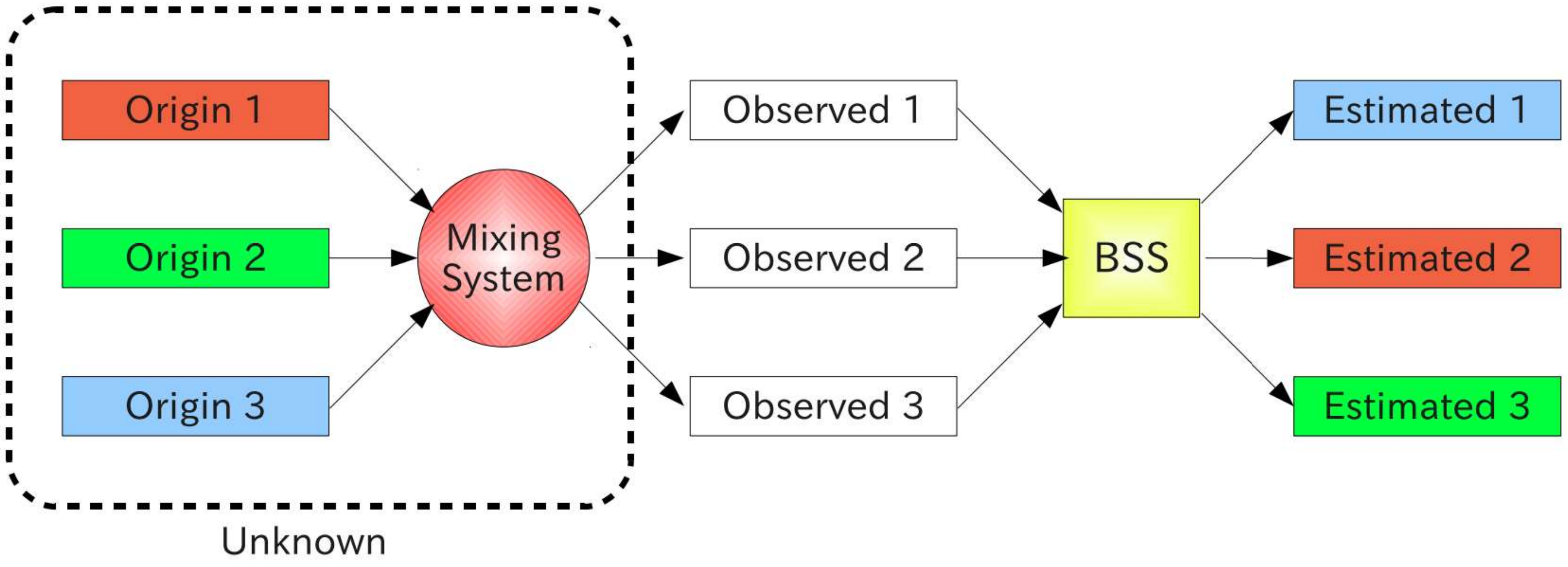
The "Cocktail Party" Problem

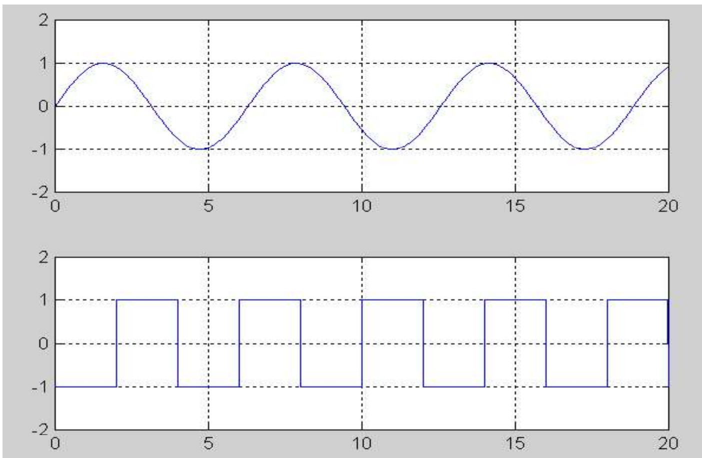


$\mathbf{x} = \mathbf{A}\mathbf{s}$

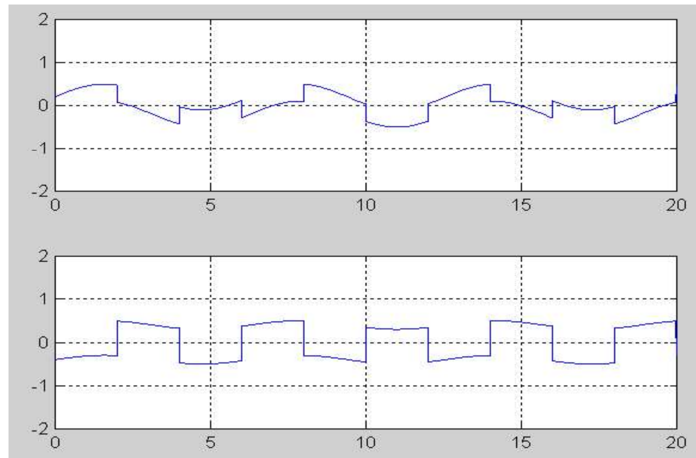
n sources, m=*n* observations

Blind source separation





Two Independent Sources



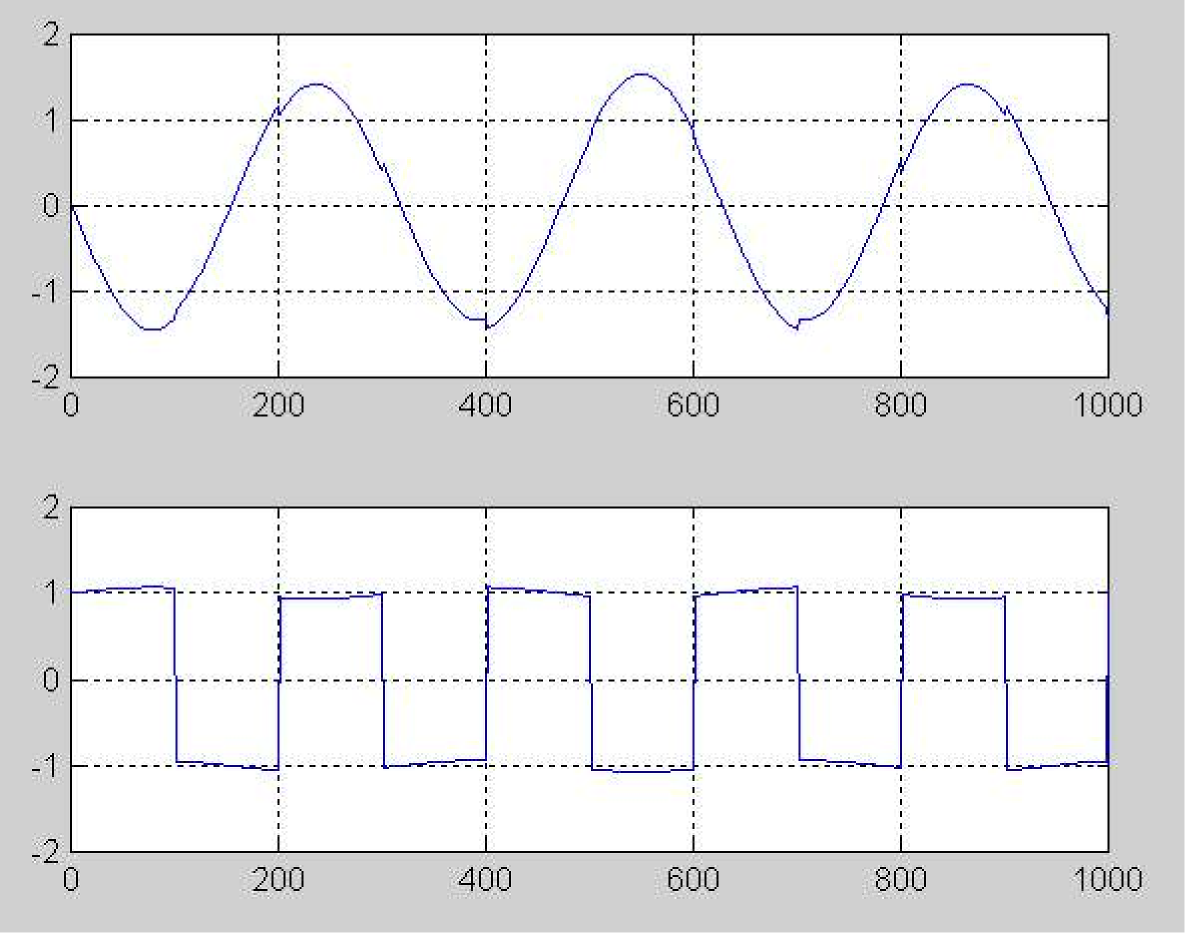
Mixture at two Mics

$$x_1(t) = a_{11}s_1 + a_{12}s_2$$

$$x_2(t) = a_{21}s_1 + a_{22}s_2$$

a_{IJ} Depend on the distances of the microphones from the speakers

Goal



Get the Independent Signals out of the Mixture

ICA Model

- $x_j = a_{j1}s_1 + a_{j2}s_2 + .. + a_{jn}s_n$, for all j

$$\mathbf{x} = \mathbf{A}\mathbf{s}$$

- IC's **s** are latent variables & are unknown AND Mixing matrix **A** is also unknown

Task: estimate **A** and **s** using only the observeable random vector **x**

- assume that no. of IC's = no of observable mixtures

  and **A** is square and invertible

- So after estimating **A**, we can compute $\mathbf{W}=\mathbf{A}^{-1}$ and hence

$$\mathbf{s} = \mathbf{W}\mathbf{x} = \mathbf{A}^{-1}\mathbf{x}$$

When can the ICA model be estimated?

Must assume:

- The si’s are mutually statistically independent
- The si’s are non-Gaussian
- (Optional:) Number of independent components is equal to number of observed variables

- Fortunately, signals measured by sensors are usually quite non-Gaussian

- Then: the mixing matrix and components can be identified [3]
    - A very surprising result!

[3] P. Comon, “Independent component analysis, A new concept?”, Signal Processing, Volume 36, Issue 3, 1994, Pages 287-314, ISSN 0165-1684

How is non-Gaussianity used in ICA ?

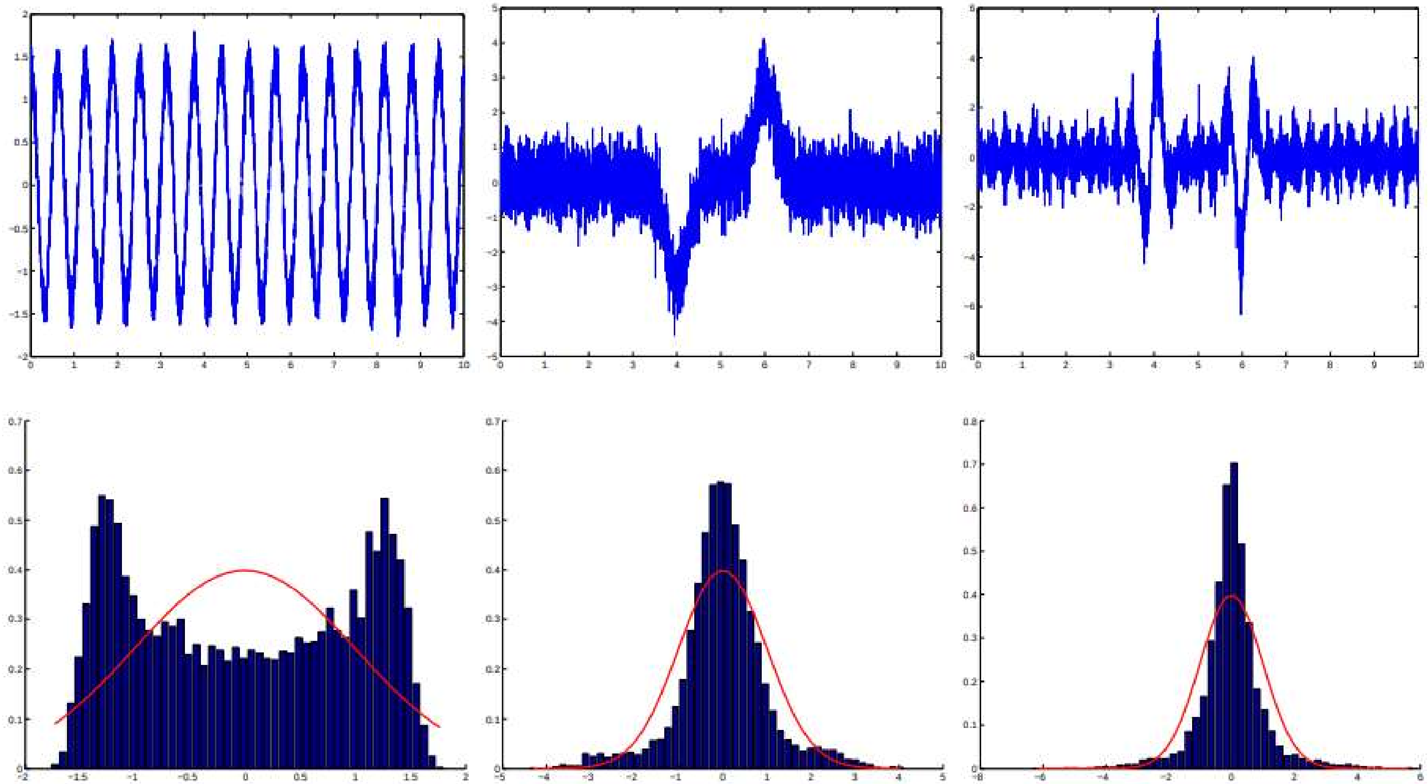
<u>Classic Central Limit Theorem:</u>

Average of many independent random variables will have a distribution that is close(r) to Gaussian

- So, roughly: any mixture of components will be more Gaussian than the components themselves

- Maximising the non-Gaussianity of $\sum w_i x_i$, we can find s_i

Sample non-Gaussian signals



The majority of ICA algorithms include two ingredients:

1. Non-Gaussianity measure
   - Kurtosis: a classical measure, but sensitive to outliers
   - Differential entropy: statistically better, difficult to compute
   - Approximations of entropy: good compromise

2. Optimisation algorithm
   - Gradient methods
   - Fast fixed-point algorithm (FastICA)[4]

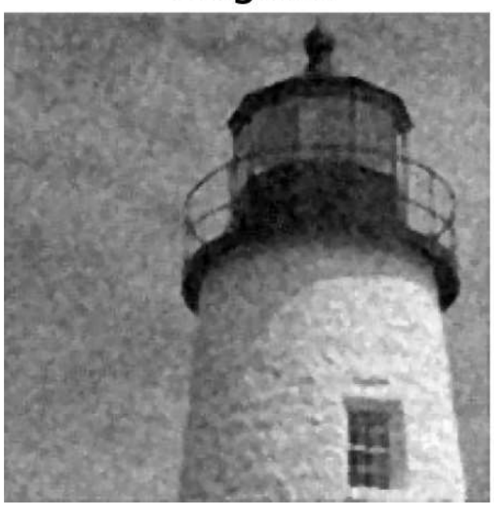
[4] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," in IEEE Transactions on Neural Networks, vol. 10, no. 3, pp. 626-634, May 1999.

ICA for Image Denoising

original

noisy

Wiener filtered



median filtered

ICA denoised

(Hoyer, Hyvarinen)