GE461

Introduction to Data Science

Data Pre-processing

Shervin R. Arashloo

Spring 2025

Summary:

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

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 used for modeling stage is acceptable and of sufficient quality

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

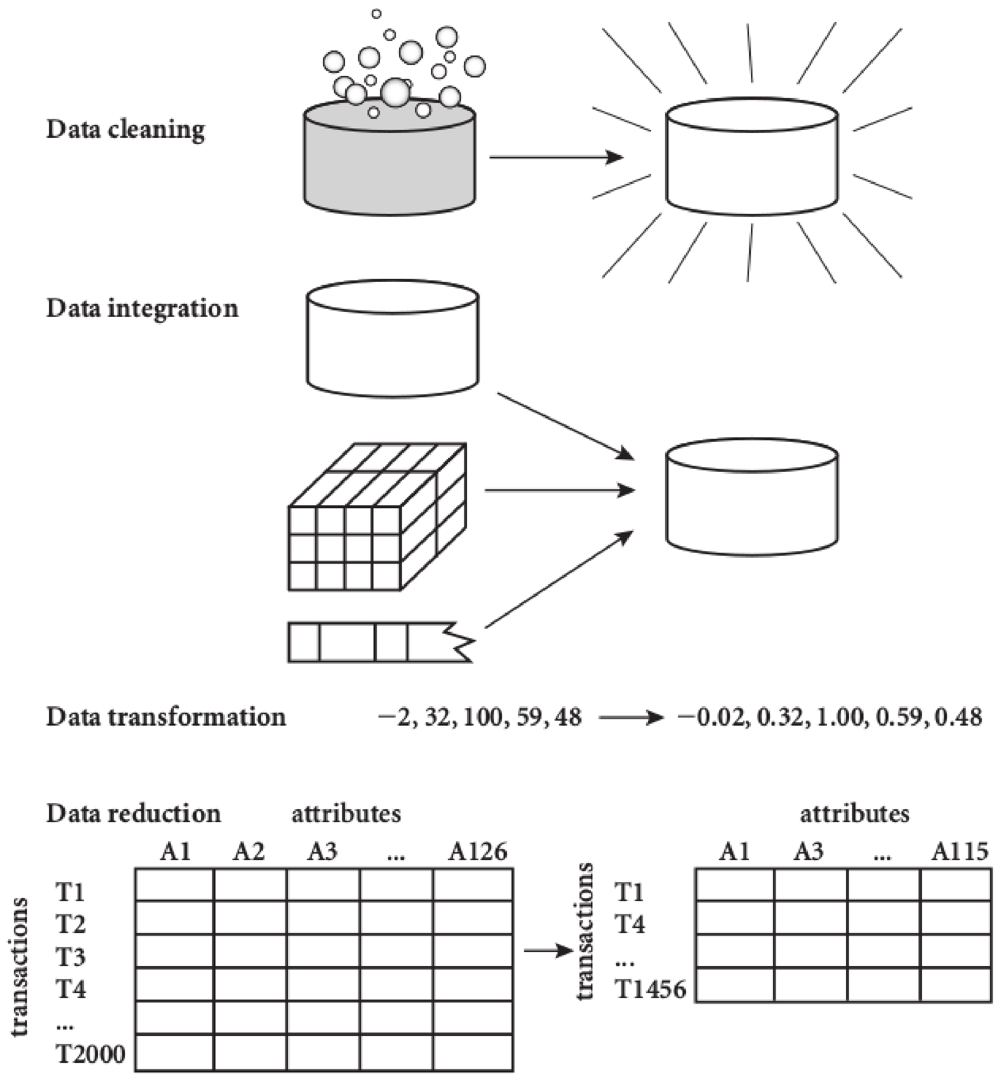
- Data preprocessing includes cleaning, instance selection, normalisation, transformation, feature extraction and selection, etc.

What Are the Benefits?

- Good data preparation is crucial to producing valid and reliable models that have high accuracy and efficiency
- It is essential to spot data issues early to avoid getting misleading predictions
- Accuracy of any analytical model depends highly on the quality of data fed into it
- High quality data leads to more useful insights which enhance organisational decision making and improve overall operational efficiency
- Data preparation conducted cautiously and with analytical mindset can save lots of time and effort, 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

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

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

- An additional shift and rescaling as

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

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

- **Transformation to a Uniform random variable**

Given a random variable x with the 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:

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

- **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

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

$$\Rightarrow \delta_x = \hat{\mu} + 2.4\hat{\sigma}.$$

- ***Lognormal distribution***

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

- The parameter estimates are: $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \log x_i \quad \text{and} \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(\log x_i - \hat{\mu})^2$

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

$$
\begin{aligned}
P(x \leqslant \delta_x) &= P(\log x \leqslant \log \delta_x) \\
&= P\left(\frac{\log x - \hat{\mu}}{\hat{\sigma}} \leqslant \frac{\log \delta_x - \hat{\mu}}{\hat{\sigma}}\right) = 0.99 \\
\Rightarrow \delta_x &= e^{\hat{\mu} + 2.4\hat{\sigma}}.
\end{aligned}
$$

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 cancel 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:
  - Spatial/temporal 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

Common types of noise:

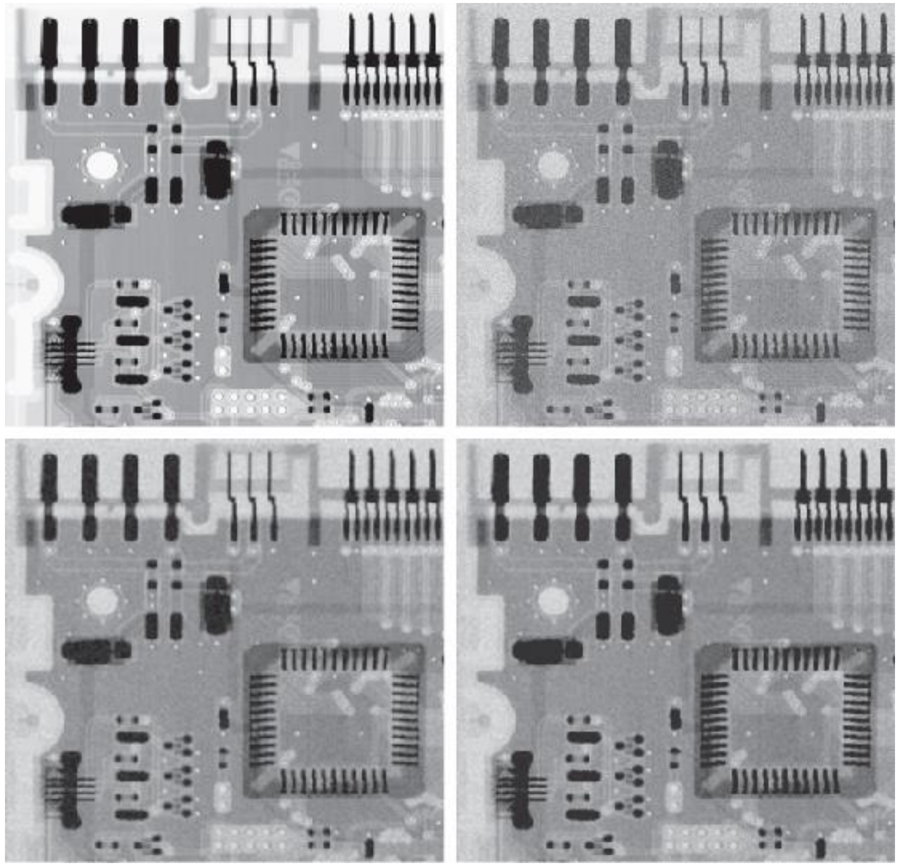
- Salt-and-pepper noise: contains random occurrences of black and white pixels.
- Impulse noise: contains random occurrences of white pixels.
- Gaussian noise: variations in intensity drawn from a Gaussian/Normal distribution.

Original

Salt and pepper noise

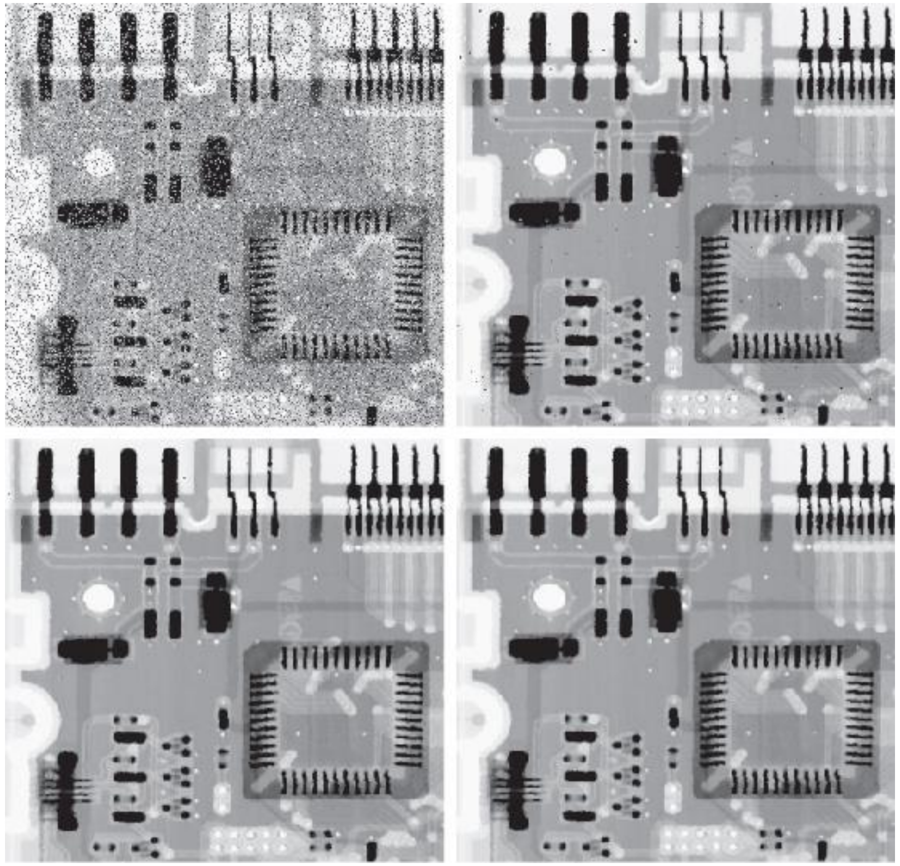
Impulse noise

Gaussian noise



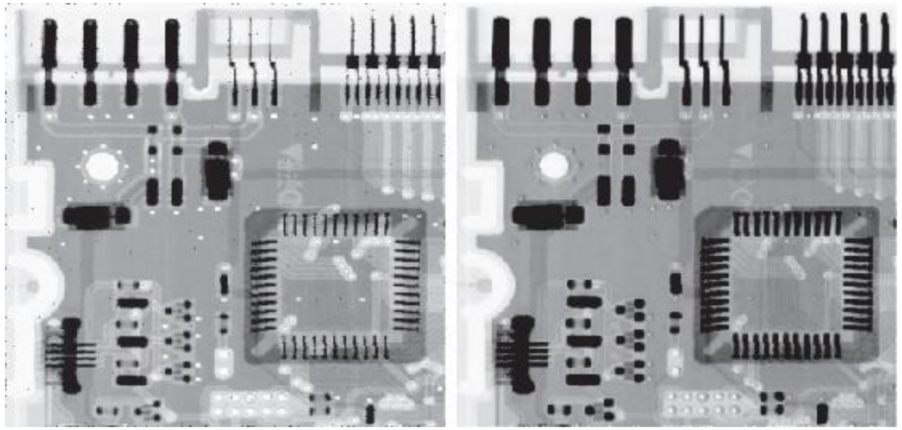
a b
c d

FIGURE 5.7 (a) X-ray image of circuit board. (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 $P_s = P_p = 0.1$. (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

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

Frequency domain filtering

- **The 2D Discrete Fourier Transform (DFT):** Fast changes in the original signal appear as high frequency components while slow changes in the signal appear as low frequency components

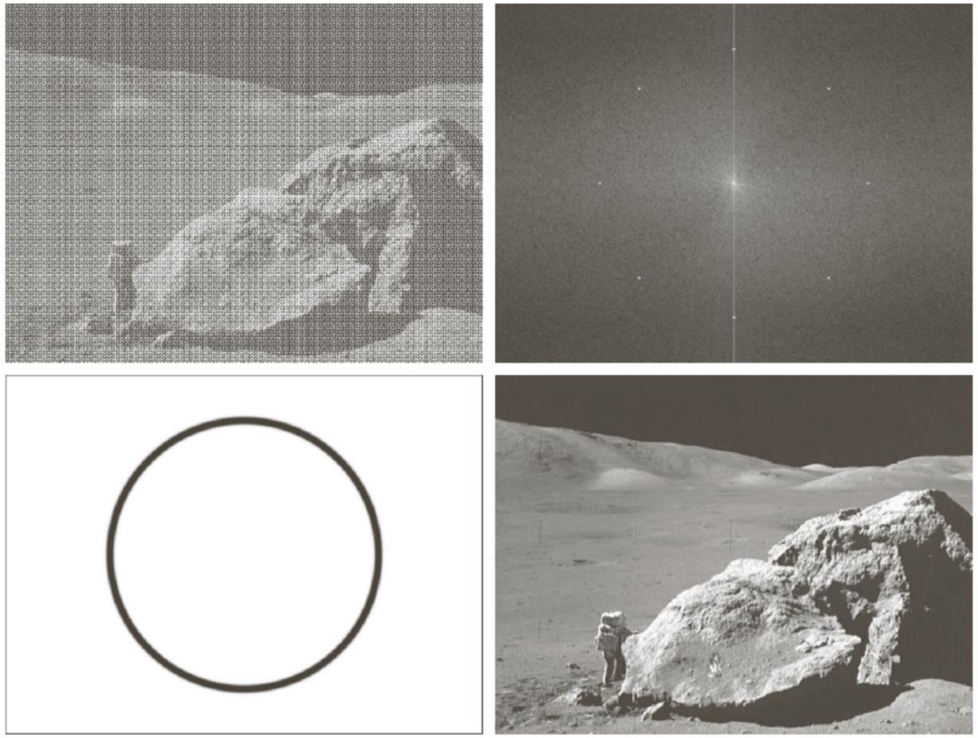
Defined for a sampled image *f(x, y)* of MxN pixels:

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

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**!

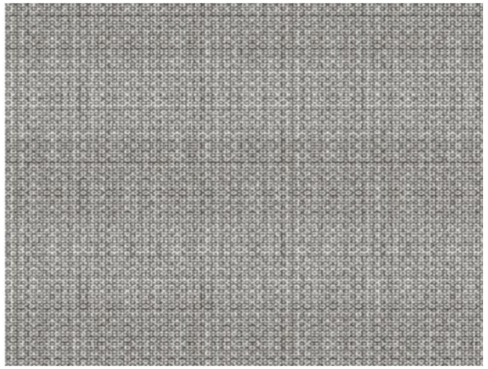
$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$$



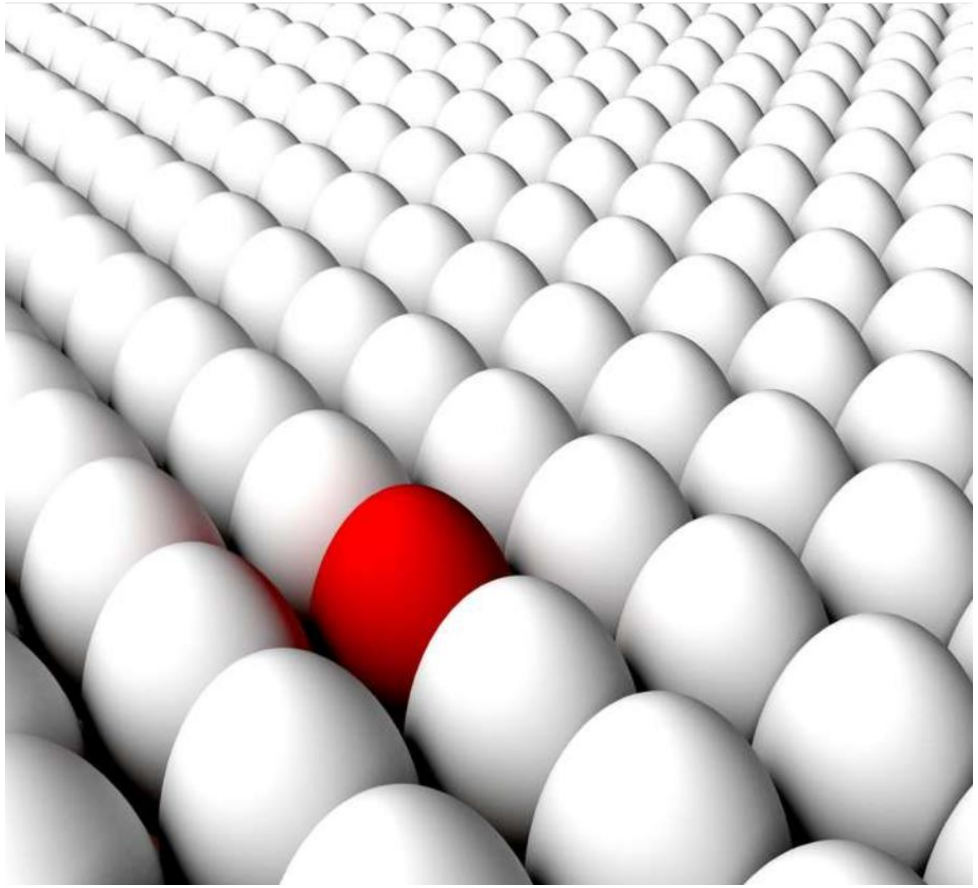
a b
c d

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 →



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 of outlier detection**

  – Fraud detection

  – Detecting measurement errors

  – Public health

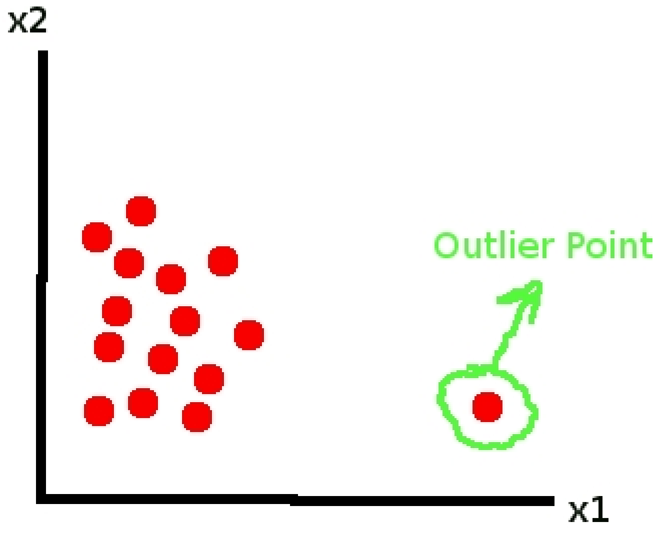
  - Medical analysis
  - Sports statistics
  - etc.

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

Types of Outliers/Anomalies

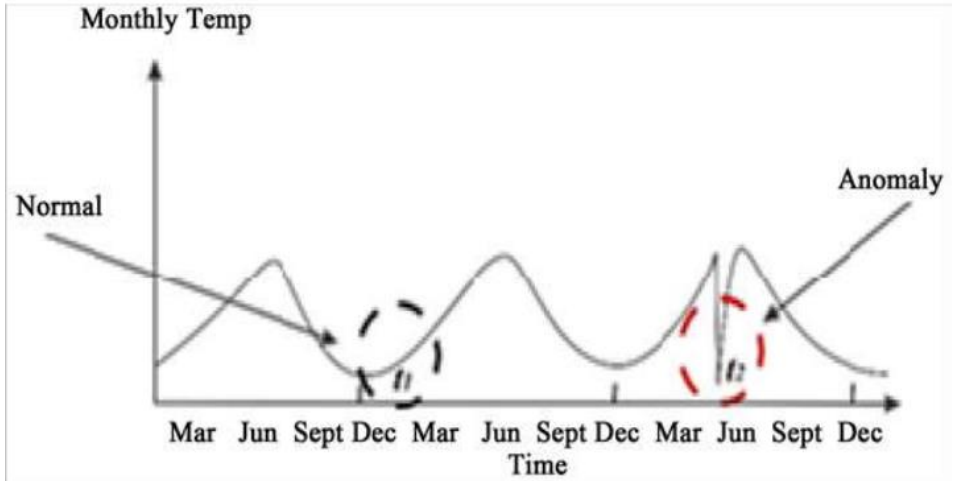
- **Point Anomaly**
  - An object that significantly deviates from the rest of the data set
  - Example: Intrusion in computer networks

- **Contextual Anomaly**
  - An object that deviates significantly based on a selected context
  - Example: temperature in a particular month



Point anomaly

Contextual anomaly



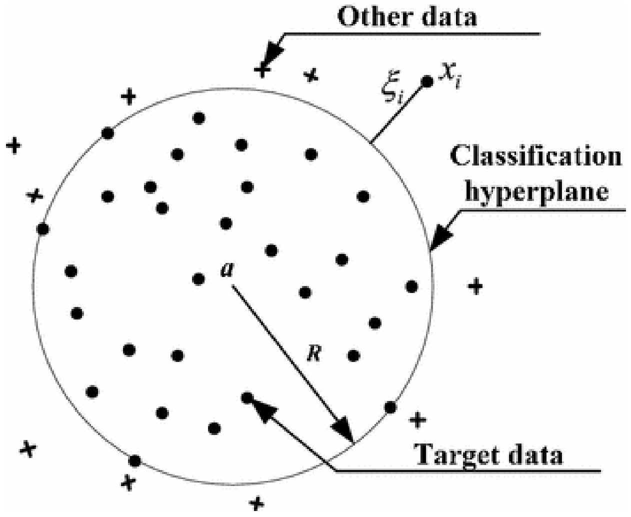
How to detect anomalies?

Different anomaly detection 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

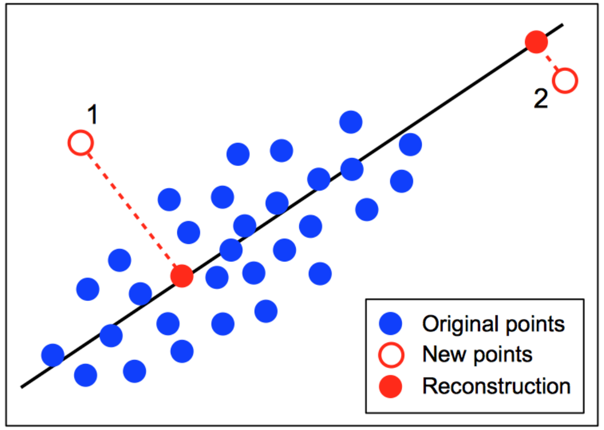
- Different OCC techniques:
  - Boundary methods
  - Reconstruction-based methods
  - Density-based methods

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

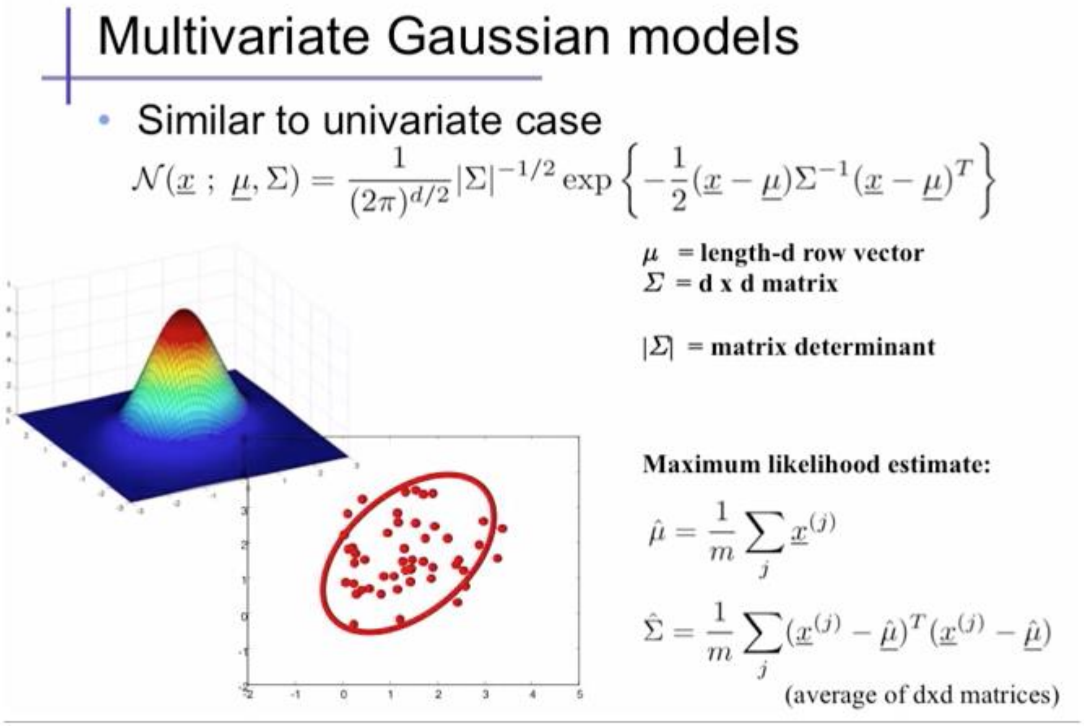


[2] Tax, D.M., Duin, R.P. Support Vector Data Description. Machine Learning 54, 45–66 (2004)

- In the <u>reconstruction-based</u> 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 shortly)

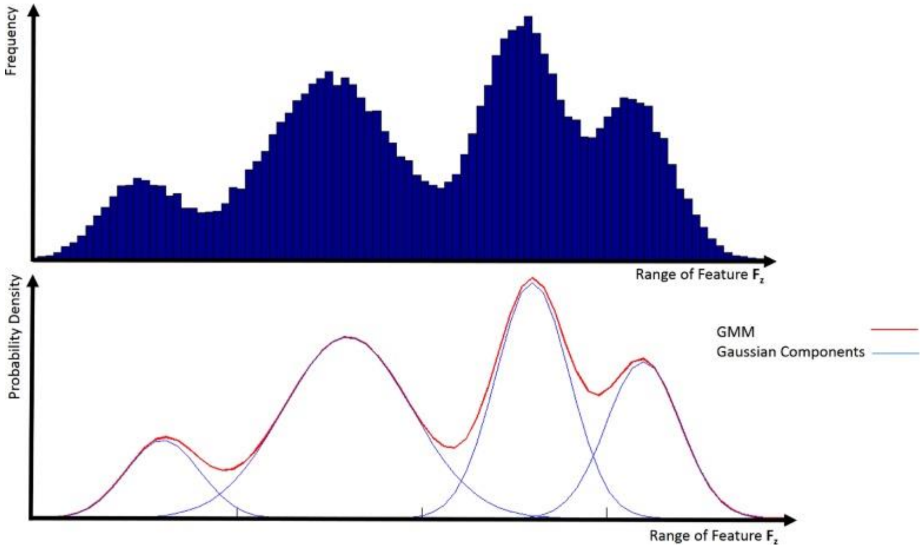


- The <u>density-based approaches</u> try to estimate the density of the training data followed by setting a threshold on the estimated density
- Several different distributions have been assumed in practice, including the Gaussian or a Poisson distribution



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



Data Reduction: Principal Component Analysis

- Given $\mathbf{x_1}, \ldots, \mathbf{x_n} \in \mathrm{R}^d$, the goal is to find a d'-dimensional subspace where the reconstruction error of $\mathbf{x_i}$ in this subspace is minimised.
- The criterion function for the reconstruction error can be defined in the least-squares sense as

$$J_{d'} = \sum_{i=1}^{n} \left\| \sum_{k=1}^{d'} \mathbf{y}_{ik} \mathbf{e_k} - \mathbf{x_i} \right\|^2$$

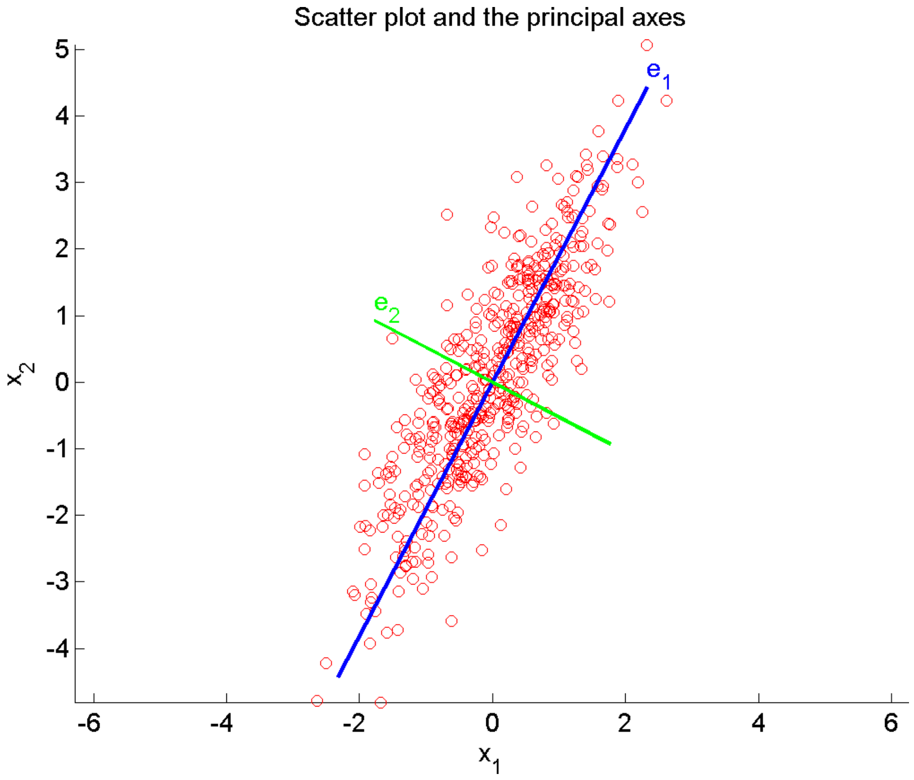
- where $\mathbf{e_1}, \ldots, \mathbf{e_{d'}}$ are the bases for the subspace (stored as the columns of $\mathbf{A}$) and $\mathbf{y_i}$ is the projection of $\mathbf{x_i}$ onto that subspace.

- It can be shown that $J_{d'}$ is minimised when $\mathbf{e_1}, \ldots, \mathbf{e_{d'}}$ are the d' eigenvectors of the *scatter matrix*

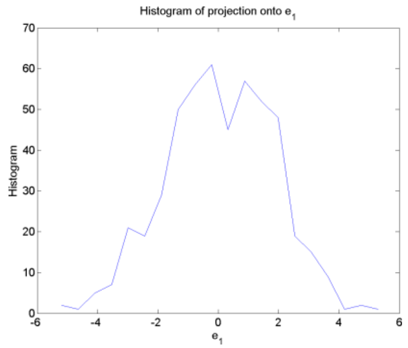
$$S = \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$$

having the largest eigenvalues.

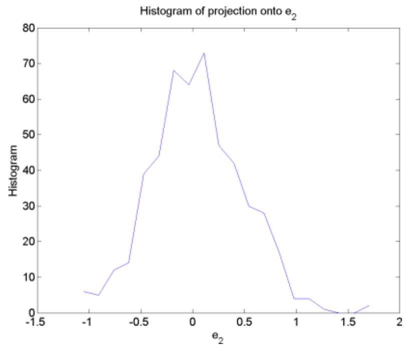
- ▶ The coefficients $\mathbf{y} = (\mathbf{y}_i, \ldots, \mathbf{y}_{d'})^T$ are called the *principal components*.
- ▶ When the eigenvectors are sorted in descending order of the corresponding eigenvalues, the greatest variance of the data lies on the first principal component, the second greatest variance on the second component, etc.
- ▶ Often there will be just a few large eigenvalues, and this implies that the d'-dimensional subspace contains the signal and the remaining $d - d'$ dimensions generally contain noise.



(a) Scatter plot.



(b) Projection onto e_1.



(c) Projection onto e_2.

Scatter plot (red dots) and the principal axes for a bivariate sample. The blue line shows the axis e_1 with the greatest variance and the green line shows the axis e_2 with the smallest variance. Features are now uncorrelated.

Example Application: Image compression

- 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 → 60D

PCA compression: 144D → 16D

PCA compression: 144D → 6D

PCA compression: 144D → 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*." [3]

[3] Hyvärinen, Aapo, and Erkki Oja. "Independent component analysis: algorithms and applications."
Neural networks 13, no. 4-5 (2000): 411-430

- 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.

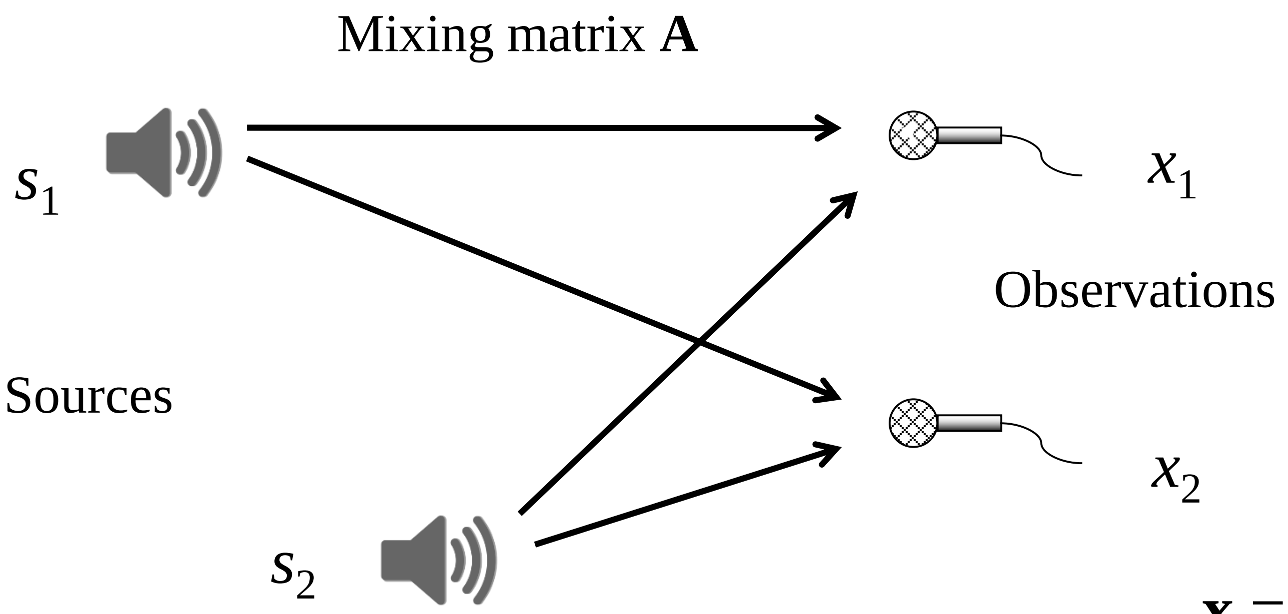
ICA seeks directions that are as **independent** from each other as possible

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

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

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

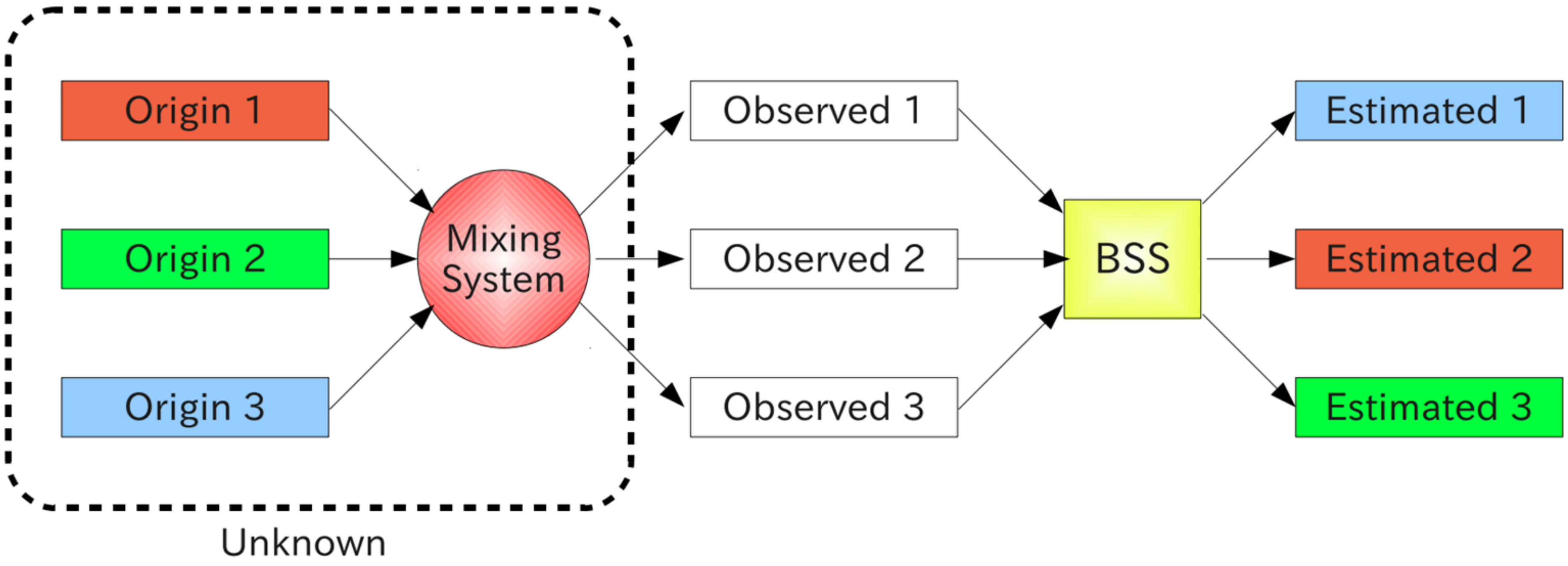
The "Cocktail Party" Problem



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

n sources, m = *n* observations

Blind Source Separation



ICA Model

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

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

- ICs "**s**" are latent variables & are unknown <u>AND</u> Mixing matrix **A** is also unknown

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

- Lets assume that no. of ICs = 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

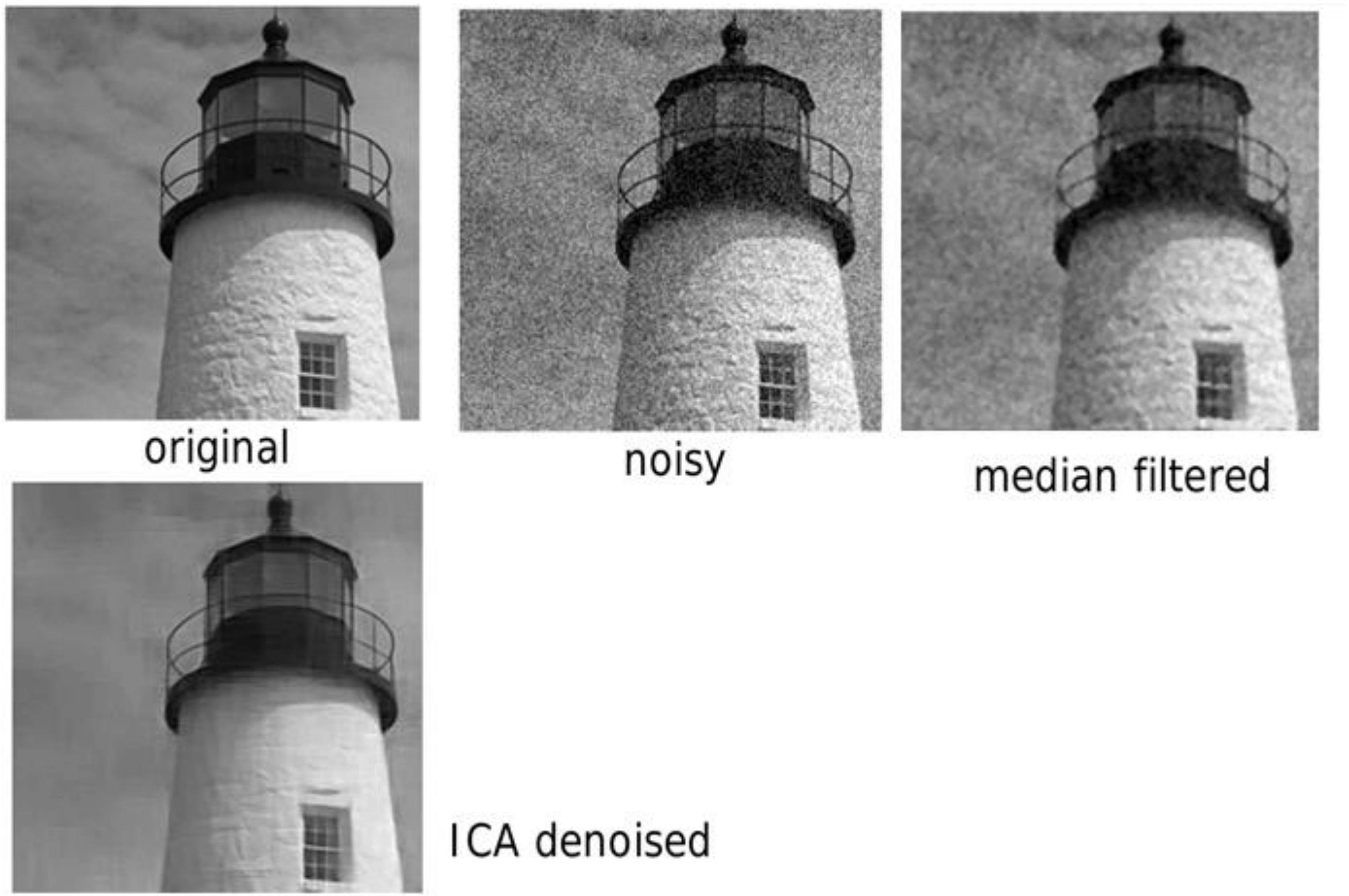
– The si's are non-Gaussian

– (Optional:) Number of independent components is equal to number of observed variables

Then: mixing matrix and components can be identified [4]. A very surprising result!

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

Example: ICA for Image Denoising



Feature Selection for Data Reduction

- The objective in many data analysis tasks is to learn a function that relates values of features to values of outcome variable(s)

–Often, we are presented with many features

–Not all of these features are informative

- Feature Selection is the task of identifying an “optimal” (take this in lay language) set of features that are useful for accurately predicting the outcome variable

Motivation for Feature Selection

- **Accuracy**

–Eliminating irrelevant features may help learn better predictive models by reducing confusion

- **Generalisability**

–Models with less features have lower complexity, so they are less prone to overfitting

- **Interpretability**

–Identifying a small set of features can help understand the mechanics of the relationship between the features and the outcome variable(s)

- **Efficiency**

–with smaller number of features, learning and prediction may take less time/space

- All possible feature subsets 2^N combinations.
- If you fix the feature subset size to M $\binom{N}{M}$
- This number of combinations is unfeasible, even for moderate M
- A search strategy is therefore needed to direct the feature selection process as it explores the space of all possible combination of features

Main Approaches

- **Treat feature selection as a separate task**

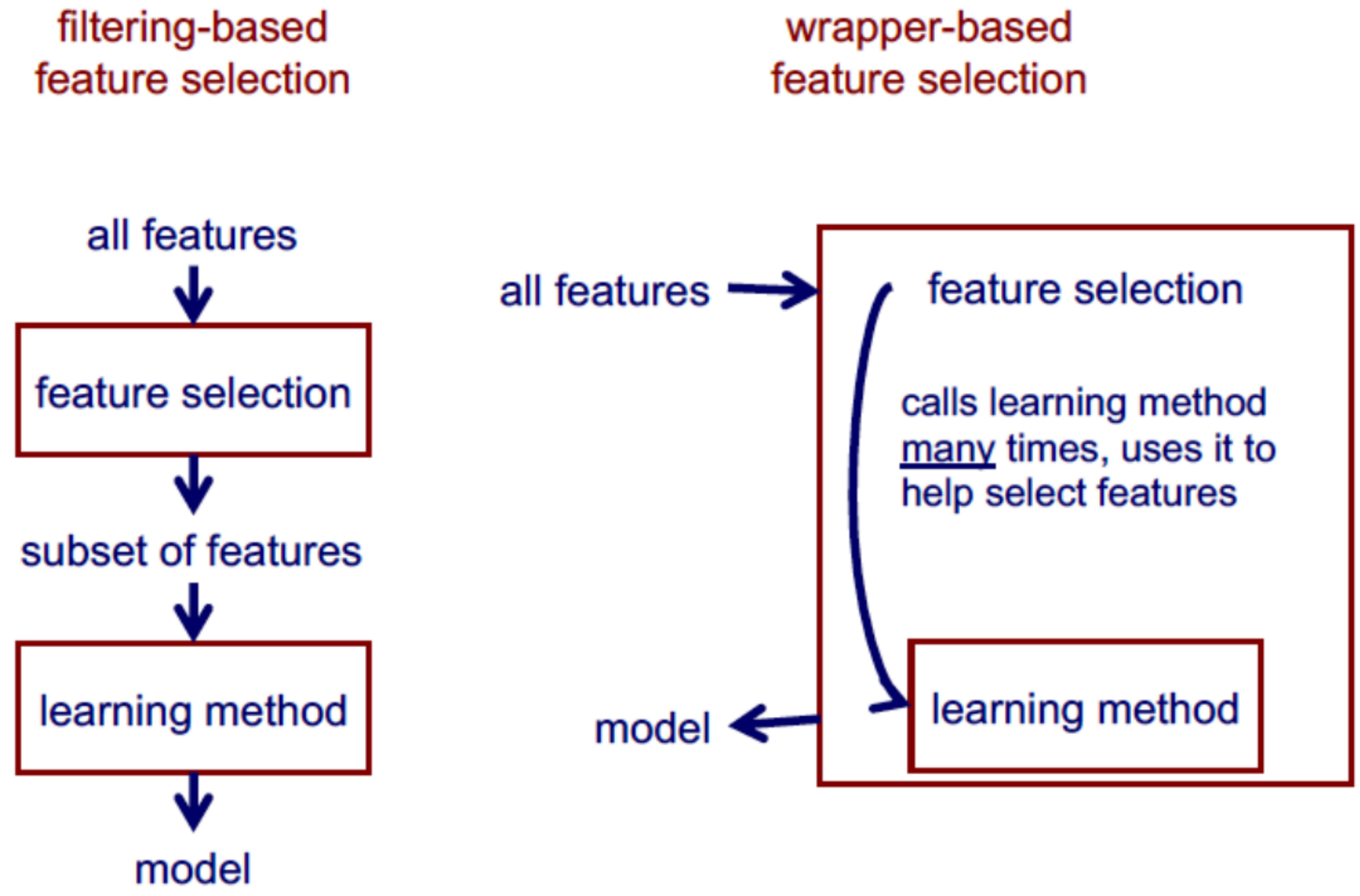
– Filtering-based feature selection

– Wrapper-based feature selection

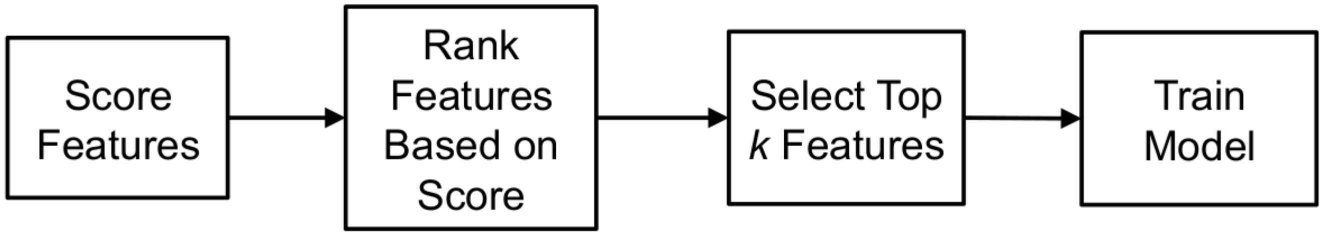
- **Embed feature selection into the task of learning a model**

– Regularisation

Feature Selection as a Separate Task



Filtering-based feature selection



- Scores do not represent prediction performance since no validation is done at this stage
- Do NOT use validation/ test samples to compute score

- *k* can be chosen heuristically
- Standard rules of thumb can be used to set a threshold (e.g., use features with *statistically significant* scores)
- Can use cross-validation to select an optimal value of *k* (using prediction performance as the criterion)

Wrapper-based feature selection

- Frame the feature selection task as a search problem
- Evaluate each feature set by using the prediction performance of the learning algorithm on that feature set
  - Cross-validation
- How to search the exponential space of feature sets?

Searching for Feature Sets

state = set of features
start state = *empty* (forward selection)
or *full* (backward elimination)

operators
add/subtract a feature

scoring function
cross-validation accuracy using learning method on a given state's feature set

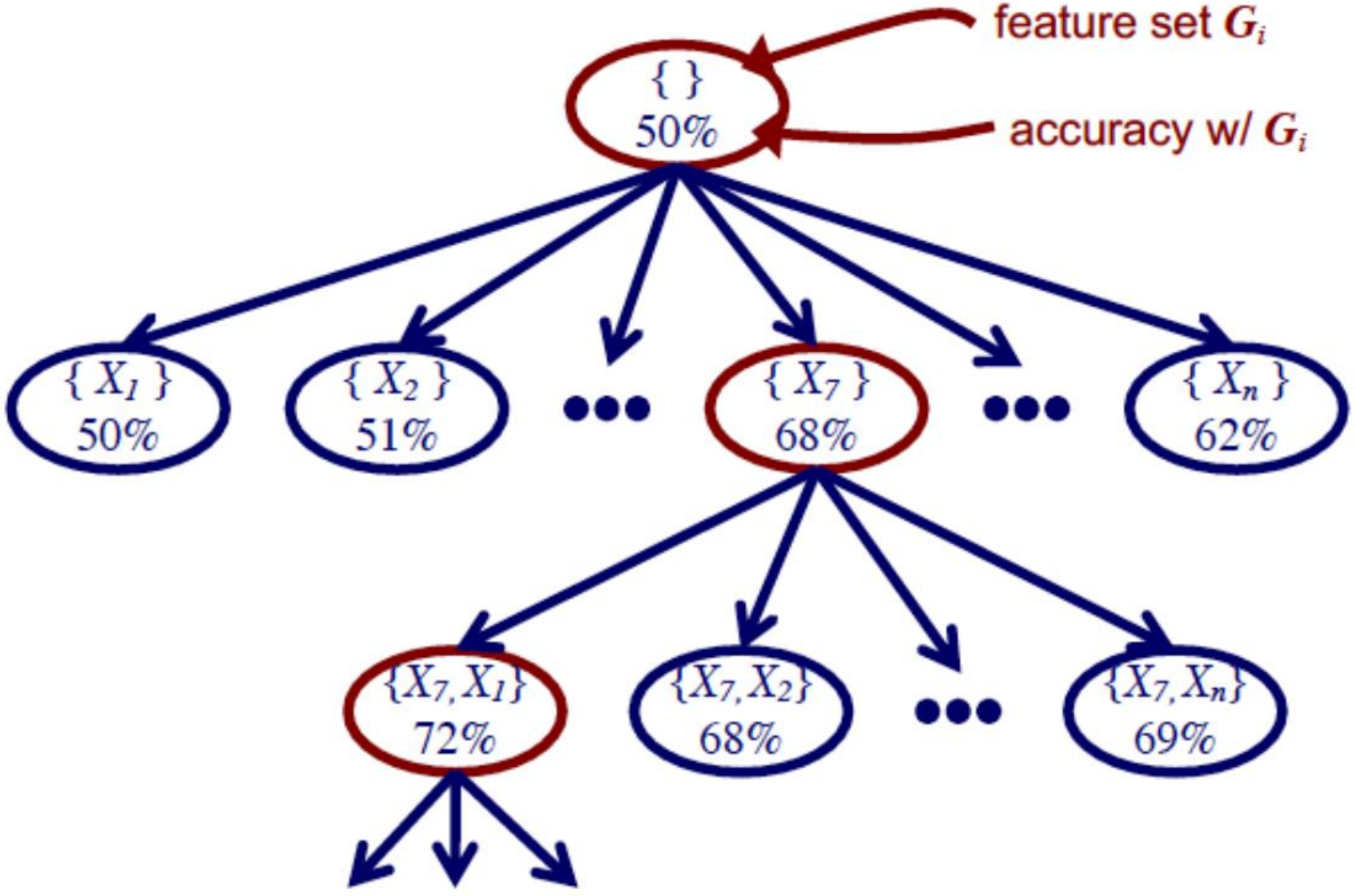
Forward Selection

Given: feature set $\{X_i, \ldots, X_n\}$, training set D, learning method L

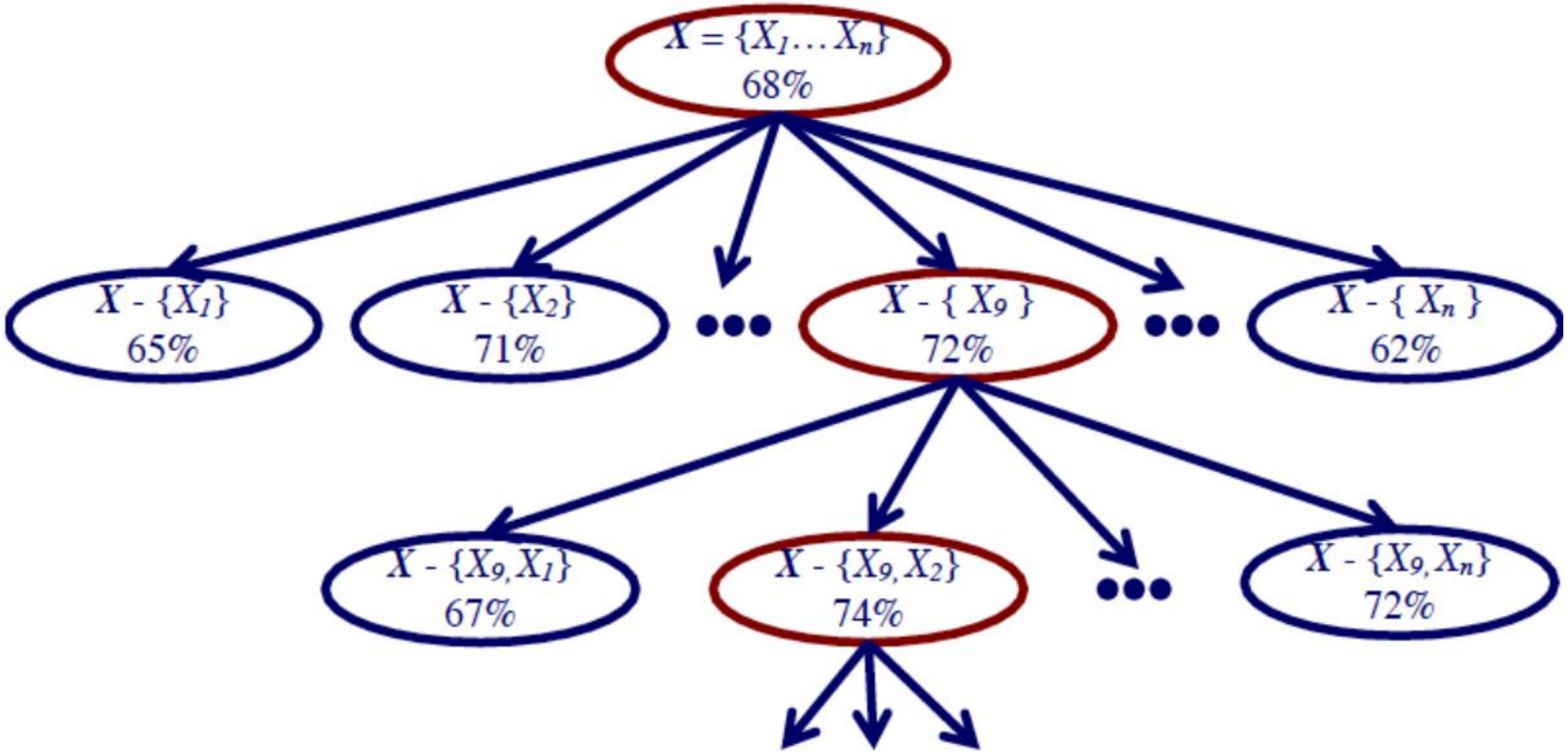
$\boldsymbol{F} \leftarrow \{\ \}$
while score of $\boldsymbol{F}$ is improving
    for $i \leftarrow 1$ to n do
        if $X_i \notin \boldsymbol{F}$
            $\boldsymbol{G}_i \leftarrow \boldsymbol{F} \cup \{ X_i \}$
            $Score_i =$ Evaluate$(\boldsymbol{G}_i, L, D)$
    $\boldsymbol{F} \leftarrow \boldsymbol{G}_b$ with best $Score_b$
return feature set $\boldsymbol{F}$

scores feature set G by learning model(s) with L and assessing its (their) accuracy

Forward Selection



Backward Elimination



Forward Selection vs. Backward Elimination

- both use a hill-climbing search

forward selection

- efficient for choosing a small subset of the features
- misses features whose usefulness requires other features (feature synergy)

backward elimination

- efficient for discarding a small subset of the features
- preserves features whose usefulness requires other features

Embedded Methods (Regularisation)

- Instead of explicitly selecting features, bias the learning process towards using a small number of features

- Key idea: objective function has two parts:
  - A term representing error minimization (model fit)
  - A term that “shrinks” parameters toward 0

LASSO

Linear regression:

$$f(\boldsymbol{x}) = w_0 + \sum_{i=1}^{n} x_i w_i$$

$$E(\boldsymbol{w}) = \sum_{d \in D} \left(y^{(d)} - f\left(\boldsymbol{x}^{(d)}\right)\right)^2$$

$$= \sum_{d \in D} \left(y^{(d)} - w_0 - \sum_{i=1}^{n} x_i^{(d)} w_i\right)^2$$

We would like to force some coefficients to be set to 0

Add L1 norm of the coefficients as the penalty term:

$$E(\boldsymbol{w}) = \sum_{d \in D} \left(y^{(d)} - w_0 - \sum_{i=1}^{n} x_i^{(d)} w_i\right)^2 + \lambda \sum_{i=1}^{n} |w_i|$$

Why does this result in more coefficients to be set to 0, effectively performing feature selection?