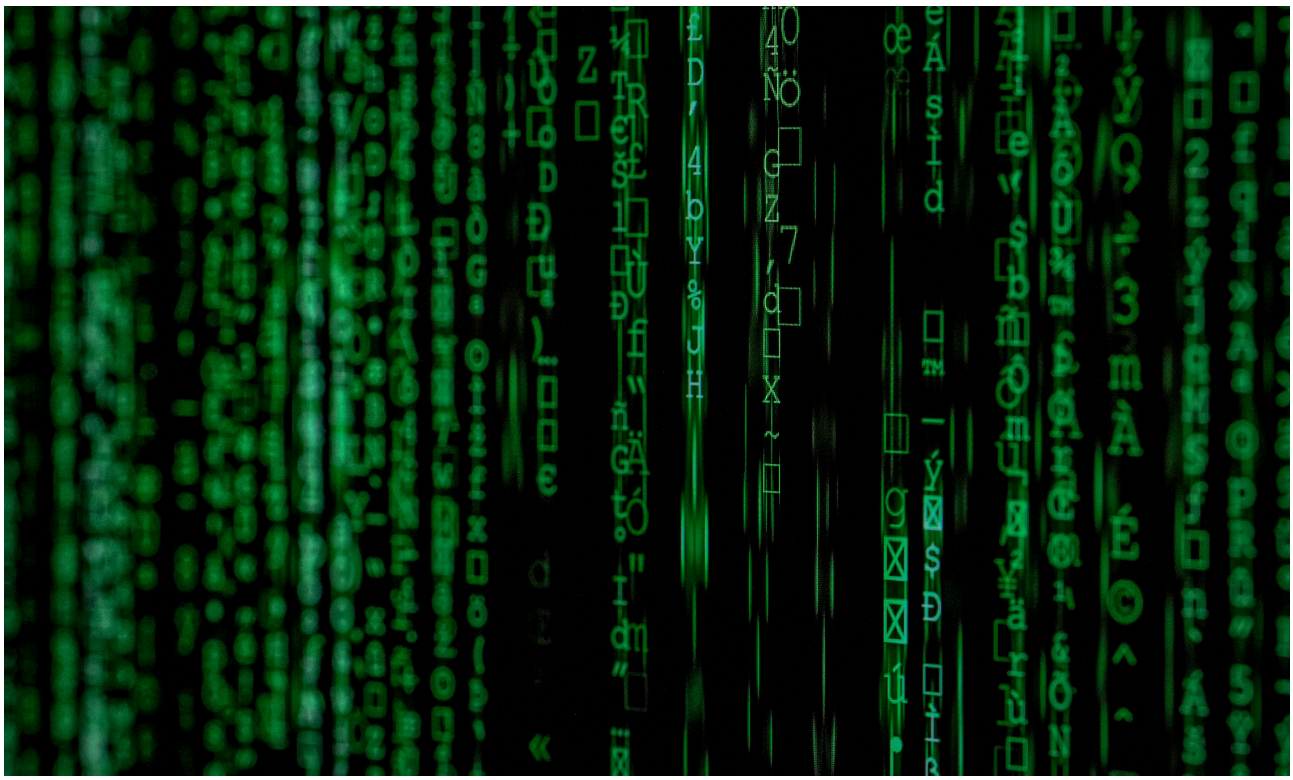
Data Mining



Notes1

Chapter 1

1. Introduction to Data Mining

Data Mining is the process of extracting knowledge from data

Data Analysis: The process of extracting knowledge from data and analyzing it

---

Machine learning: A science field concerned with the development of algorithms for computer to process, analyze and learn from data

Statistics: A subfield of mathematics containing various tools for trend analysis, probabilistic predictions and solid theoretical bases to build various models.

Data analysys tasks

Prediction Tasks: Using known variables and patterns to predict unknown and future data → Regression, Time Series prediction, Classification.

Description Tasks: Finding patterns in a data set can be understood by a human → Clustering, Association rules mining, Sequential pattern discovery.

Example: rule mining

Given a set of records, rule mining consists in producing dependency rules which will predict the occurrence of an item

ID	Items
1	bread, coke, milk
2	beer, bread
3	beer, coke, diaper, milk
4	beer, bread, coke, diaper, milk
5	coke, bread, milk

Rule mined:
$\{milk\} \rightarrow \{coke\}$
$\{diaper, milk\} \rightarrow \{beer\}$

Steps of Knowledge Discovery from Data process

- Learning the application domain (the features)
  - It’s mandatory to do a good analysis
- Creating a target: data selection
  - only use samples of your data (70% OF YOUR DATA)
- **Data cleaning** and pre-processing
  - If you do not do that well your data will be byased
- **Data reduction and transformation**
  - Find usally features, dimensionality/variable reduction, invariant representation
- Choosing a data mining task
  - Summarization, classification, regression
- Choosing a **Machine learning Algorithm**
- **Results evaluation and knowledge presentation**

GOALS:

ISSUE AND CHALLENGES:

Make data intelligible	Human interaction
Retrive and select patterns	Data pre processing: outlayers, noisy data
Determine the validity of the extracted information	Large Dataset, overfitting…

2. What are data?

A **data** (datum) is a basic description of a thing, an individual, a fact, an instruction or a phenomenon.

- A data is made of one or several descriptive criteria caled variables or features:
  - A single feature: Univariate
  - Multiple features: Multivariate

A data set is a set containing several data, usually identified by an id. The id is not considered a variable.

→ A data set can be rapresented into a matrix (N lines for N elements in our dataset, D columns for D features)

Type of variables

- **Quantitative**
  - Conuntuons
  - Discrete

- **Qualitative**

Are the categorical variables: Binary data, coulors etc

- ordered (bad, medium, good)
- unordered (eye color)

Dealing with missing data

There are 2 ways of dealing with missing values:

1. Ignoring the tuples with missing values
2. Fill the missing values when feasible
   a. fill with the mean
   b. Deduce the missing value from similar data
   c. Use the most probable value for the missing attribute (Bayesian inference)

Outliers

An outlier is an *aberrant value* corresponding to a bad measure, a miscalculation, a mistake or misrepresentation

There are several ways to deal with them:

- Ignoring them or deleting them, *if* their numbers are not too high and their distribution sufficiently random.
- Keeping them and tolerating a small margin of error.
- Treat them as missing values and replace them

---

3. Univariate statistics of variables

Univariate statistics of variables are applicable to **data sets with a single variable**, or to individual variables in a data set with several features.

They are data that have only one variable, we can only have that feature in order to descrive your data

- Extracting mean, range, variance, distribution
- Summarize content using graphs
- Detecting anomalies

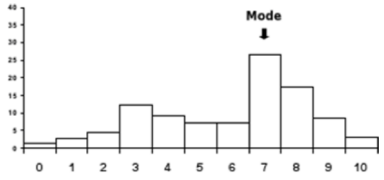
Measures of central tendency

A measure of central tendency is a single value that attempts to describe a set of data by identifying the central position within this set. As such, measures of central tendency are sometimes called measures of central location. They are also classified as summary statistics.

The Mode

The mode(s) si the most frequently occurring value(s) in the data set.

- The mode is most easily identified in an ordered frequency histogram.
- On a histogram it represents the highest bar.
- The mode is not necessarily unique.
- In surveys, the mode is the most popular option.



The mode can be used with categorical unordered data.

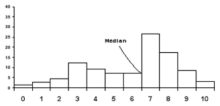
Modes can also be found in data distributions. A distribution can be Unimodal, Bimodal, Multimodal

The Median

The median is a measure that allows to define the value which cuts the distribution in *two equal parts*.

- In an ordered data set, the median is computed as follows:

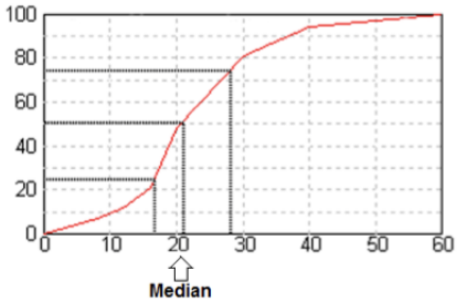
$$median = \begin{cases} x_{\frac{N+1}{2}} & \text{if } N \text{ is odd} \\ \frac{1}{2}\left(x_{\frac{N}{2}} + x_{\frac{N+1}{2}}\right) & \text{if } N \text{ is even} \end{cases}$$



→ The data set needs to be sorted from lowest to highest values in order to compute the median.

The median is the value which cuts an ordered set in two subsets of the same size

If you have even data, your median will. be the mean of the 2 value in the middle.



Advantages:

- The median si often used when there are extreme values that could greatly influence the mean and distort what might be considered typical.
- This often is the case with home prices and with income data for a group of people, which are often very skewed. For such data, the median often is reported instead of the mean.

- For example, ni a group of people where the salary of one person is 10 times the mean, the mean salary of the group may be unusualy dragged up. In this case, the median may better represent the typical salary level of the group.

The Mean

The mean is equal to the sum of all the values in the data set divided by the number of values in the dataset:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Disadvantages

The mean is sensitive to extreme values

Dispersion

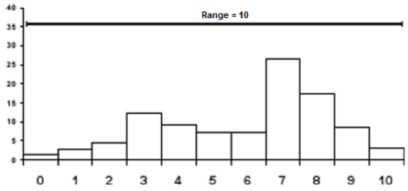
A dispersion criterion is a value which represents the homogeneity in the values of a variable.

- **Statistical dispersion** (also called statistical variability or variation) is the variability is a variable or a probability distribution.
- A measure of statistical dispersion si a NON-NEGATIVE real number equal to zero if al the data are the same and that increases as the data become more diverse:
  - STD deviation
  - Median absolute deviation
  - Mean absolute deviation

Range

The range is the difference between the largest and the smallest observed value

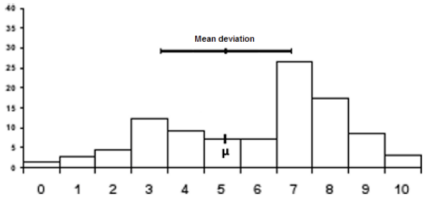
$$Range = X_{max} - X_{min}$$



Mean Deviation

The man deviation is the avarage deviation from the mean value

$$\sigma_{\bar{x}} = \frac{1}{N}\sum_{i=1}^{N} |x_i - \bar{x}|$$



Variance

is the mean of the squared deviation of a variable from its expected value or mean.

$$\sigma_X^2 = var(X) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

Standard Deviation

The standard deviation describes the "typical" difference between the observations and the mean value

$$s_X = \sqrt{s^2} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

Sample	Mean	Standard deviation
79,80,81	80	$\sqrt{\frac{2}{3}}$
60,80,100	80	$20\sqrt{\frac{2}{3}}$

Data Distribution

A distribution describes the likelihood (probability) for a variable to take a given value.

The describe a unimodal distribution, we analyse:

- its mean
- its stand deviations

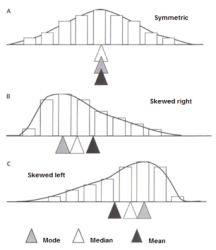
To further analyze the distribution, one may also see its shape using

- Its degree of **Skewness**
- Its degree of **Kurtosis**

Skewness

- Symmetric when:
  $mode = median = mean$
- Skewed right when:
  $mode < median < mean$
- Skewed left when:
  $mode > median > mean$

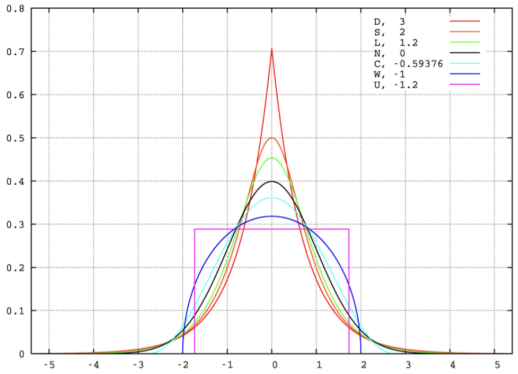
The skewness of a distribution can be determined as follows:
- Symmetric when:
  $mode = median = mean$
- Skewed right when:
  $mode < median < mean$
- Skewed left when:
  $mode > median > mean$



Kurtosis

$$\tilde{Ku}_X = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^4}{N \times s^4(X)}$$

- When $Ku_X < 0$, the distribution is **Leptokurtic**.
- When $Ku_X > 0$, the distribution is **Platykurtic**.



4. Bivariate Data

Bivariate data can be stored in a table with two columns:

	X	Y
obs 1	2	1
obs 2	3	5

Example: Temperature (X) and precipitation (Y) are measured on a given day at a set of weather stations.

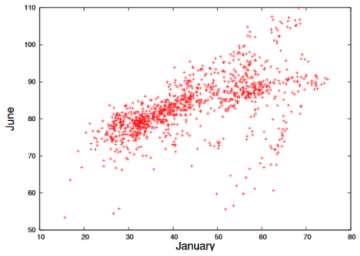
Analyzing bivariate data

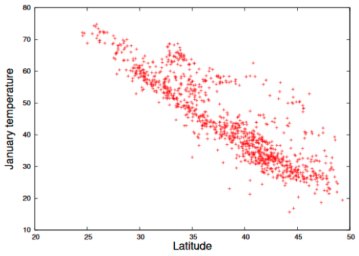
When the data are described by two o random variables (e.g. X and Y), we are interested in knowing the possible statistical link between these two variables.

Scatterplot

The most important graphical summary of bivariate data si the scatterplot. It is simply a plot of the points (Xi, Yi) in the plane.

The folowing figures show scatterplots of June maximum temperatures against January maximum temperatures, and of January maximum temperatures against latitude. Al temperatures are in degree Fahrenheit.





A key feature to look for in a scatterplot si the association, or trend between X and Y.

- Higher January temperatures tend to be paired with higher June temperatures, so these two values have a positive association
- Higher latitudes tend to be paired with lower January temperature decreases, so these values have a negative association.

Remark: If higher X values are paired with low or with high Y values equally often, there is no association.

— It si **important** to remember that it is ill advised to draw causal implications from statements about associations, unless your data come from a randomized experiment. —

→ Just because January and June temperatures increase together does not mean that January temperature cause June temperature to increase (and vice versa).

In general, if X and Y have an association, then:

- X could cause Y to change
- Y could cause X to change
- An external variable Z (perhaps unknown) could cause both X and Y to change.

Unless your data come from a randomized experiment, statistical analysis alone is not capable of answering questions about causality.

Covariance

The covariance between 2 random variables assess the joint difference between their respective means values.

The covariance denoted with Cov(X,Y) is:

Regular Covariance

$$Cov(X,Y) = \frac{1}{n}\sum_{i=1}^{N}(x_i - \mu_X)(y_i - \mu_Y)$$

Covariance of a sample

$$Cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{N}(x_i - m_X)(y_i - m_Y)$$

Properties:

$$Cov(X,Y) = Cov(Y,X)$$
$$Cov(X,X) = VAR(X) = \sigma_X^2$$
$$Cov(a \times X, b \times Y) = a \times b \times Cov(X,Y)$$
$$Cov(a + X, b + Y) = Cov(X,Y)$$

- **When two variables are fully independent, their covariance is null. However, the opposite is not always true!**
- **The covariance is very sensitive to the unit and scale of the observed variables!**
- The covariance is often difficult to interpret.

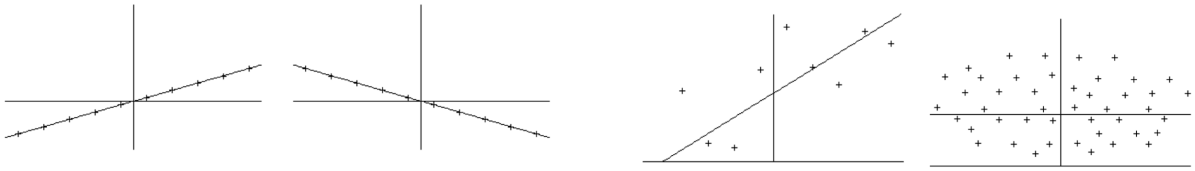
We need to find more accurate and easier to use criteria

Correlation Coefficient

The Pearson Product-moment correlation coefficient is a measure of the linear correlation between two random variables.

$$r = Cor(X,Y) = \frac{Cov(X,Y)}{s_x s_y}$$

- The correlation coefficient takes values between 1- and +1 inclusive.
- +1 denotes a complete correlation.
- 0 denotes no correlation between the two variables.
- -1 means a total negative correlation.



(a) $r = 1$ (b) $r = -1$ (c) $r = 0.77$ (d) $r \approx 0$

WARNING: **Correlation does not imply causality!**

Example: The number of sunburn as a function of the number of person. The interpretation of the correlation coefficient is sometimes counter-intuitive. Example: si r = -0.6 si strong correlation ?

Better coefficient: Coefficient of determination

Coefficient of determination

The coefficient of determination si the proportion of the variance of Y, which disappears if Xsi fixed (or the other way around).

- It is the square value of the correlation coefficient.
- r^2 is a proportion between 0 and 1, and is very easy to interpret.

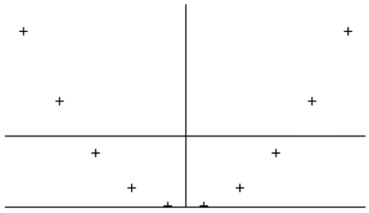
$$r^2 = \left(\frac{Cov(X, Y)}{s_x s_y} \right)^2$$

- r2 = 0.36: it means that 36% of Y information si contained in X
- It's not bad, but ti also means that 64% of Y information cannot be deduced from X.

ACTUNG:

Do not confuse correlation and causality: A strong correlation between two variables can reveal a causal relationship between them, but not necessarily.

→ In the example there is clearly correlation



LIMITS:

A single outlying value can have a substantial effect on the correlation coefficient

CONFIDENCE INTERVAL FOR THE CORRELATION COEFFICIENT:

IC_{95} of r and r^2

- If X (resp. Y) is fixed, and the distribution of Y (resp. X) follows a normal distribution (often difficult to assess):
  - Then $Z = \frac{\ln(1+r) - \ln(1-r)}{2}$ follows a normal distribution
  - With $s_Z = \sqrt{1/(n-3)}$,
  - Then $Z_{inf} = Z - 1.96 s_Z$, $Z_{sup} = Z + 1.96 s_Z$,
  - And $IC_{95}(r) = \left[\frac{e^{2Z_{inf}} - 1}{e^{2Z_{inf}} + 1}, \frac{e^{2Z_{sup}} - 1}{e^{2Z_{sup}} + 1} \right]$
- **Otherwise: The bootstrap method still works !**
- For the coefficient of determination: $IC_{95}(r^2) = (IC_{95}(r))^2$

5. Mining bivariate categorical data

So far we have been interested in finding correlations between quantitative ideally continuous variables

- What about categorical variables?
- How to find whether there is a correlation between two categorical variables: hair color and eye color, neighbourhood and type of job, etc.

The correlations methods described before cannot be applied.

Categorical Variable: Chi square test

Chi-squared test of independence: χ^2

- The Chi-squared is based on the **contingency table** (cross table) of the possibles values of two variables.
- It is computed based on the difference between the **expected frequencies** and the **observed frequencies** of one or more categories of the contingency table.
- A zero Chi-squared means that the two variables are completely independent.
- A non-zero Chi-squared is more difficult to interpret:
  - Requires to evaluated the likelihood of the resulting Chi-squared based on its known distribution. (Requires tables or a calculator)
  - Can be evaluated using the Chuprov contingency coefficient or Cramer's V.

- Consider two variables i and j so that i€[1,…,r] and j€[1,…,c].
- Let o_{i,j} be the number of observed data so that the first feature's value is i and the second feature's value is j
- Let n_i be the total number of elements having i as a value for there first feature.
- Let N be the total number of observations.

Then, the expected contingency e_{ij} computes as follows:

$$e_{i,j} = \frac{n_i \cdot n_j}{N}$$

Computing the χ^2

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(o_{i,j} - e_{i,j})^2}{e_{i,j}}$$

EXAMPLE:

	Dark hair	Light hair
Brown eyes	32	12
Blue eyes	14	22
Green eyes	6	9

Notations examples

$i \in$ (Dark hair, light hair) $\quad j \in$ (Brown eyes, Blue eyes, Green eyes)

$o_{dark,brown} = 32 \quad e_{dark,brown} = \frac{44 \times 52}{95} = 24.08$

Hypothesis Test

In statistics, **p-values** are criteria often linked to what si called a hypothesis test.

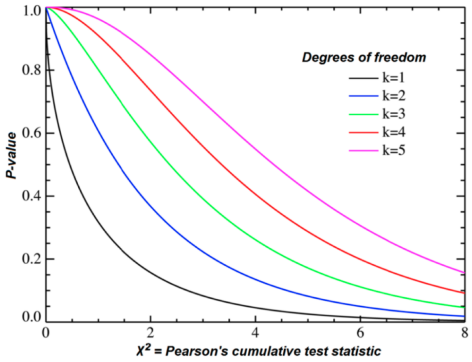
Hypothesis Test

- State your **null hypothesis** H_0 and alternative hypotheses.
- Choose a value α (usually 0.1, 0.05 or 0.01)
- Compute the p-value for your proposed model:
  - if p-value $< \alpha$: reject H_0.
  - if p-value $\geq \alpha$: you cannot reject H_0.

Computing the p-value

The p-value is computed using the known distribution of the Chi-squared
function (using tables, a calculator, or a graph) considering degrees of freedom of the problem.

Degrees of freedom = (r - 1)(c -1)



A p-value close to zero rejects the hypothesis that the two variables are independent

We can say that there is “1-pvalue”% of chance that the apparent dependence between the 2 variables is not random luck.

The **Chuprov** contingency coefficient (sometimes spelled Tschuprow) can be used to interpret the result of a Chi-Squared

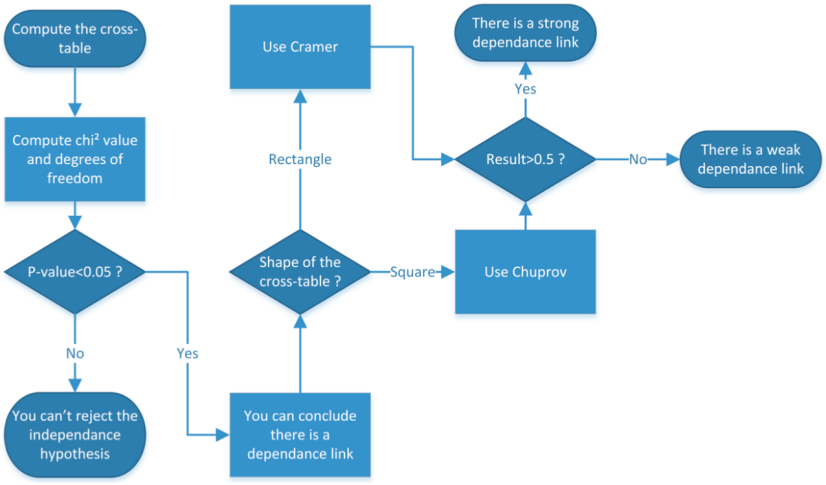
$$\rho = \sqrt{\frac{\chi^2}{N\sqrt{(c-1)(r-1)}}}$$

Cramer's V IS another index measuring the amount of dependency between two variables.

$$V = \sqrt{\frac{\chi^2}{N \cdot min(c-1, r-1)}}$$

REMARKS:

- While the p-value evaluates whether a result based on the chi-squared has good chances of being significant, its value is not proportional to the amount of correlation.
- Chuprov coefficient is best reliable with square contingency tables, while Cramer's V works best with rectangular ones.
- Both Chuprov coefficient and Cramer's V can be very biased: unevenly distributed observations, one variable with much more possible values than the other, etc.
- Both Chuprov coefficients and Cramer's V can't be used fi the result of the p-value shows that the chi-squared result si not significant.
- The Chi-squared can also be used to assess the correlation between categorical and quantitative data: the quantitative data need to be regrouped (examples with grades: grades from 0 to 5, 6 ot 10, 1 ot 15 and 16 to 20).



BACK TO THE EXAMPLE:

	Dark hair	Light hair	n_j
Brown eyes	$o_{1,1} = 32$ $e_{1,1} = 24.1$	$o_{1,2} = 12$ $e_{1,2} = 19.9$	44
Blue eyes	$o_{2,1} = 14$ $e_{2,1} = 19.7$	$o_{2,2} = 22$ $e_{2,2} = 16.3$	36
Green eyes	$o_{3,1} = 6$ $e_{3,1} = 8.2$	$o_{3,2} = 9$ $e_{3,2} = 6.8$	15
n_i	52	43	$N = 95$

$$\chi^2 = 10.67$$

$k = (3-1)(2-1) = 2$ degrees of freedom

With $\chi^2 = 10.67$, we have a p-value bellow 0.01: There is a significant correlation between hair color and eye color.

$\chi^2 = 10.67$ p-value< 0.01

$$\rho = \sqrt{\frac{\chi^2}{N\sqrt{(c-1)(r-1)}}} = \sqrt{\frac{10.67}{95\sqrt{2}}} = 0.28$$

$$V = \sqrt{\frac{\chi^2}{N \cdot min(c-1, r-1)}} = \sqrt{\frac{10.67}{95 \cdot min(1,2)}} = 0.34$$

6. Introduction to multivariate data

When each data is described by multiple values (i.e. several random variables), the intuitive analysis and visualization of the data becomes difficult or impossible.

- Some features may be redundant.
- The variables may be different in nature (numerical, categorical, binary, etc.)
- There may be missing values.

Missing Values

Missing values can be a problem because they make the data unreliable and prevent some calculations to be done. There are several ways to deal with them:

- Ignoring them when it is possible.
- Removing the data that have missing values.

• Trying to fill in the missing values.

- Finding the most similar complete data, and using its value to fill in the missing ones.
  - Weakness: Artificially increases any correlation coefficients.
- Replacing the missing values with the mean of the considered variable.
  - Weakness: Reduces dispersion criteria.
- Using regression techniques (Cf Lecture 3) to imput the missing variable from the values of the others.
  - Weakness: Artificially increases any correlation coefficients due to overfitting problems.

Dimensionality reduction

It si sometimes possible to reduce the dimensionality of a multivariate data set: fI only 2 or 3 variable remain, it si possible to visualize the data with minimal loss of information.

We will see in the next course, that Principal component analysis

Normalizing the data

1. Unit normalization: scale the data between 0 and 1.

Min-Max normalization

$$\tilde{X} = \begin{pmatrix} \frac{x_{1,1}-min(X_1)}{max(X_1)-min(X_1)} & \cdots & \frac{x_{1,D}-min(X_D)}{max(X_D)-min(X_D)} \\ \vdots & \ddots & \vdots \\ \frac{x_{N,1}-min(X_1)}{max(X_1)-min(X_1)} & \cdots & \frac{x_{N,D}-min(X_D)}{max(X_D)-min(X_D)} \end{pmatrix}$$

2. Logarithmic normalization: reduced the effect of outlayers, useful when attribute non linear correlations

logarithmic normalization

$$Y = \begin{pmatrix} \log_a(1+b\frac{x_{1,1}-min(X_1)}{max(X_1)-min(X_1)}) & \cdots & \log_a(1+b\frac{x_{1,D}-min(X_D)}{max(X_D)-min(X_D)}) \\ \vdots & \ddots & \vdots \\ \log_a(1+b\frac{x_{N,1}-min(X_1)}{max(X_1)-min(X_1)}) & \cdots & \log_a(1+b\frac{x_{N,D}-min(X_D)}{max(X_D)-min(X_D)}) \end{pmatrix}$$

3. Centered normalization: center data around zero

Centered reduced variables

$$\bar{M} = \begin{pmatrix} x_{1,1}-\mu_1 & \cdots & x_{1,D}-\mu_D \\ \vdots & \ddots & \vdots \\ x_{N,1}-\mu_1 & \cdots & x_{N,D}-\mu_D \end{pmatrix}$$

Once the data have been normalized (one way or another), regular univariate and bivariate statistics measures can be used to describe the relations between the different variables.

The Variance-Covariance matrix denoted C or I is a common correlation measure for multivariate data.

$$C = \begin{pmatrix} VAR(X_1) & \cdots & Cov(X_D, X_1) \\ \vdots & \ddots & \vdots \\ Cov(X_1, X_D) & \cdots & VAR(X_D) \end{pmatrix}$$

When the data are reduced, then the covariance is equal to the correlation and we obtain a correlation matrix:

$$\tilde{C} = \begin{pmatrix} 1 & \cdots & Cor(X_D, X_1) \\ \vdots & \ddots & \vdots \\ Cor(X_1, X_D) & \cdots & 1 \end{pmatrix}$$

Notes2

1. Introduction

Unsupervised learning is a Machine Learning task the aim of which is to find hidden structures and patterns in un-labeled data.

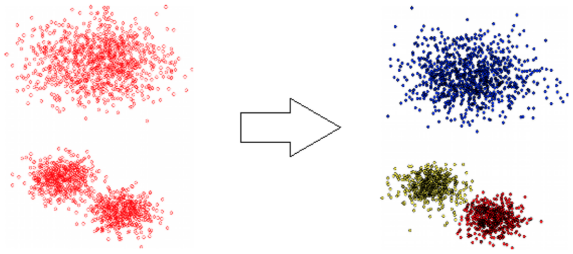
→ unsupervised learning focuses on links and similarities between the data themselves

It therefore differs from supervised learning which builds a model given already labeled data.

Data Partitioning

Data partitioning or **clustering** - is a Machine Learning task of exploratory data mining the aim of which is to split a data set made of several data (also called objects, data objects, or observations) into several subsets containing homogeneous and similar data.

The groups created via this process are called clusters.



2. Clustering

A cluster si a group of relatively homogeneous data that share more common characteristics between each other than with the data belonging to other clusters.

The notion of cluster therefore relies on the notion of *similarity* and *dissimilarity* between the data

What is similarity?

The similarity or dissimilarity between two observations si most often computed using a **distance function**. Depending on the data and the application, some distances may prove more appropriate than others

Euclidian distance	$\|\|a-b\|\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidian distance	$\|\|a-b\|\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan distance	$\|\|a-b\|\|_1 = \sum_i \|a_i - b_i\|$
Maximum distance	$\|\|a-b\|\|_\infty = \max_i \|a_i - b_i\|$
Mahalanobis distance	$\sqrt{(a-b)^\top S^{-1}(a-b)}$ where S is the covariance matrix
Hamming distance	$Hamming(a,b) = \sum_i (1 - \delta_{a_i,b_i})$

Clustering partitions

Let us denote X = {×1,..., XN}, Xn € Rˆd a data data set containing N observations described by d real features, and C = (c1,..., ck} the set of possible clusters.

The result of a clustering algorithm is called a clustering partition. Depending on whether or not a data can belong to several clusters, the partition can be **hard**, **soft**, or **fuzzy**.

(a) Hard clustering

	c_1	c_2	c_3
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1
x_4	0	0	1

(b) Soft clustering

	c_1	c_2	c_3
x_1	1	1	0
x_2	0	1	1
x_3	0	0	1
x_4	0	0	1

(c) Fuzzy clustering

	c_1	c_2	c_3
x_1	0.9	0.1	0
x_2	0	0.8	0.2
x_3	0	0.3	0.7
x_4	0	0	1.0

Let us note S the clustering partition:

- In hard clustering, S is a vector $S = \{s_1, \cdots, s_N\}, s_n \in [1..K]$
- In soft clustering, S is a matrix $S = \{s_{n,k}\}_{(N\times K)}, s_{n,k} \in (0, 1)$
- In fuzzy clustering, S is a matrix too:
  $S = \{s_{n,k}\}_{(N\times K)}, s_{n,k} \in [0, 1] \quad \forall n, \sum_k s_{n,k} = 1$

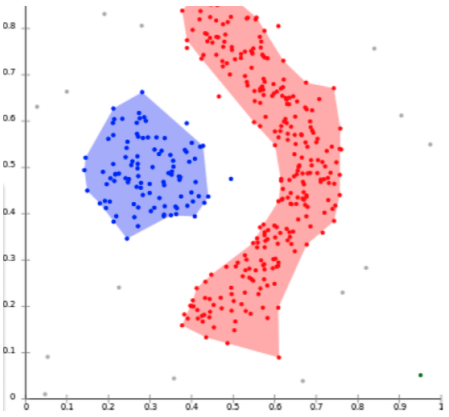
Now that we know what we want to do, the question is: Given a data set, a distance function and a desired type of partition, how do we find the clusters?

1. **Density-Based clustering algorithms**

   Inspired from the human vision: the clusters are defined as high density spaces and are separated by low density areas. (DBSCAN or OPTICS)

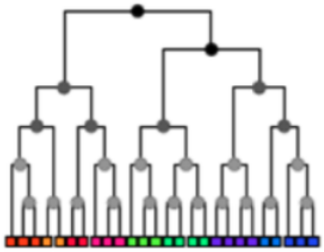
Properties

- Pros: Does not presume the shape of the clusters or their number, detect outliers.
- Cons: High complexity $O(N^2)$, difficult to parametrize



1. **Hierarchical clustering algorithms**

   - Inspired from phylogenetic trees in biology.
   - Approaches: agglomerative, divisive
   - Examples of algorithms: HCA, CURE, CLINK



Properties

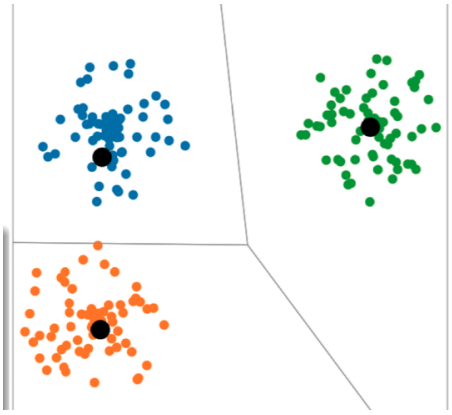
- Pros: Highlight hierarchical cluster structures, comprehensive visualization of the clusters
- Cons: High complexity $O(N^2 \log N) to O(N^3)$, choosing where to cut the dendrogram can be cumbersome.

3. **Prototype-based clustering algorithms**

- Uses vector quantization to sum up the data using representatives (e.g. mean values)
- Examples of algorithms: K-Means, Fuzzy C-Means

Properties

- Pros: Lower complexity $O(N)$
- Cons: Requires to give the number of clusters and to presume of their shapes.

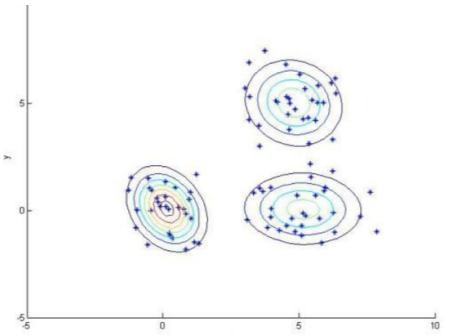


4. **Distribution-based clustering algorithms**

- Uses strong mathematical models to describe the clusters (e.g. mixture of gaussian distributions)
- Examplesofalgorithms: EM

Properties

- Pros: Lower complexity $O(N)$, strong models, convergence proofs
- Cons: Requires to give the number of clusters and to assume that the clusters will follow a given distribution.



5. **Spectral clustering algorithms**

- Sees clustering as a graph partitioning problem using the similarity matrix between the different objects.
- Examples of algorithms: Shi-Malik algorithm (normalized cuts), Meila-Shi algorithms.
- Mainly use for image analysis

Properties

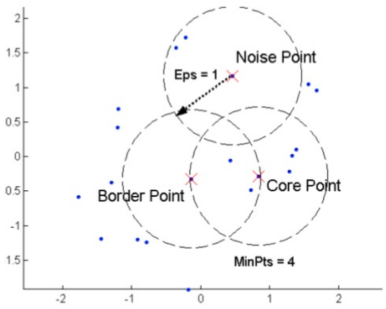
- Pros: Good results, does not assume any shape for the clusters, fast with sparse data.
- Cons: High complexity $O(N^3)$, does not scale well, not very intuitive.

---

3. Examples of clustering algorithms

DBSCAN

DBSCAN is a density based algorithm:



- The "Density" is based on the number of data within the radius "Eps" of a given element.
- A point is a **core point** if it has more data in its neighboring radius than a specified number of points "MinPts".
- Core points are the main structures for the clusters.

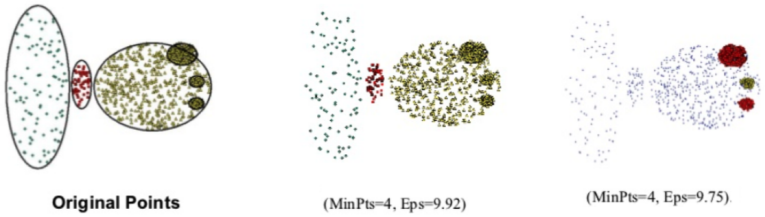
- A **noise point** has fewer points in its radius Eps than the number MinPts and is not within range of a core point.
- A **border point** has fewer points in its radius Eps than the number MinPts but it is within range of a core point.

STRENGHT

Resistant to noise and can handle clusters of different shapes and sizes

LIMITS

- Difficult to parametrise
- High Computational cost
- Does not handle well varying densities
  - OPTICS is a good evolution of DBSCAN that handles varying densities a lot better.



HCA

Hierarchical Clustering Algorithm (HCA) is a clustering algorithm inspired by phylogenetic trees in biology. It aims to group data objects based on their similarity or dissimilarity.

HCA starts with each data object as a separate cluster and then repeatedly merges the most similar clusters until a stopping criterion is met.

HCA uses a distance function (**linkage criterion**) to measure the similarity or dissimilarity between data objects.

Name	Formula	Comments
Single-linkage	$D_s(c_1, c_2) = \min_{x \in c_1, y \in c_2} d(x, y)$	Distance between the two closest elements
Complete-linkage	$D_c(c_1, c_2) = \max_{x \in c_1, y \in c_2} d(x, y)$	Distance between the two farthest element
Average-linkage	$D_a(c_1, c_2) = \frac{1}{\lvert c_1 \rvert \lvert c_2 \rvert} \sum_{x \in c_1} \sum_{y \in c_2} d(x, y)$	Average pairwise distance
Centroid-linkage	$D_\mu(c_1, c_2) = \lvert\lvert \mu_1 - \mu_2 \rvert\rvert$	Distance between the centroids

The result of the HCA algorithm is a hierarchical tree-like structure called a dendrogram, which shows the hierarchical relationships between clusters. The dendrogram can be cut at different levels to obtain different clusterings.

STRENGHT:

HCA has strengths such as its ability to handle clusters of different shapes and sizes and its resistance to noise.

LIMITS:

However, it can be difficult to parameterize and has a high computational cost. For datasets with varying densities, OPTICS (Ordering Points To Identify the Clustering Structure) is a recommended alternative to HCA.

DIFFERENT LINKAGES

Average-linkage

- Less noise sensitive, tends to favor hyper-spherical clusters
- High computational cost for roughly the same results as the centroid-linkage.

Centroid-linkage

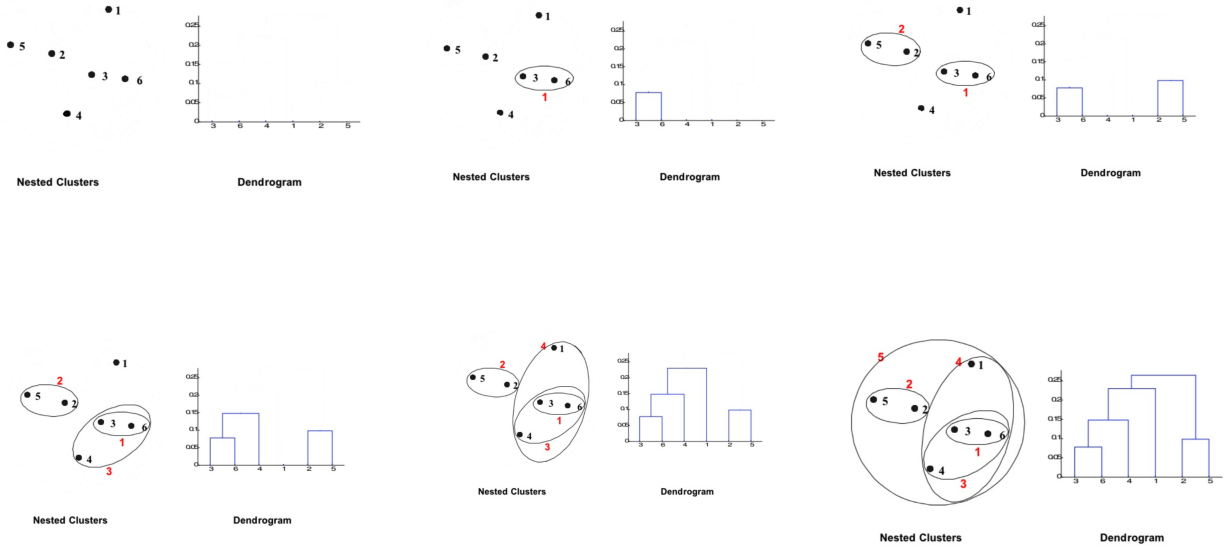
- A good compromise between single and complete link.
- Less sensitive to noise and outliers.
- tends to favor elliptical clusters.

Ward's method

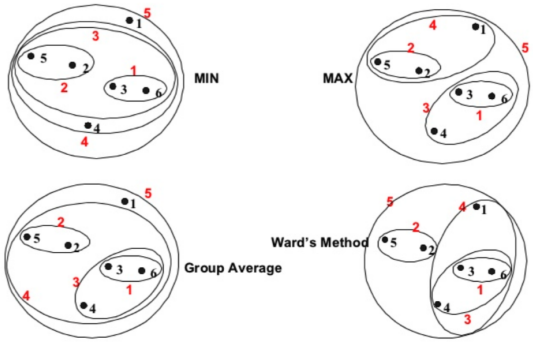
This similarity is based on the increase on the mean squared distance to the centroid when 2 clusters are merged.

- Less sensitive to noise and outliers.
- tends to favor elliptical clusters.

EXAMPLE:



On the left an example of Different Methods



K-Means

The method of the K Means, or K-Means algorithm, is a special case of the mobile centers methods. Its principle consists in using a certain number of representatives (centroids, or prototypes) in the data space. Each of these representative will represent a cluster.

- In the end, each data is associated to its closest prototype in order to obtain the clustering partition
- The groups formed with this process are homogeneous and well separated

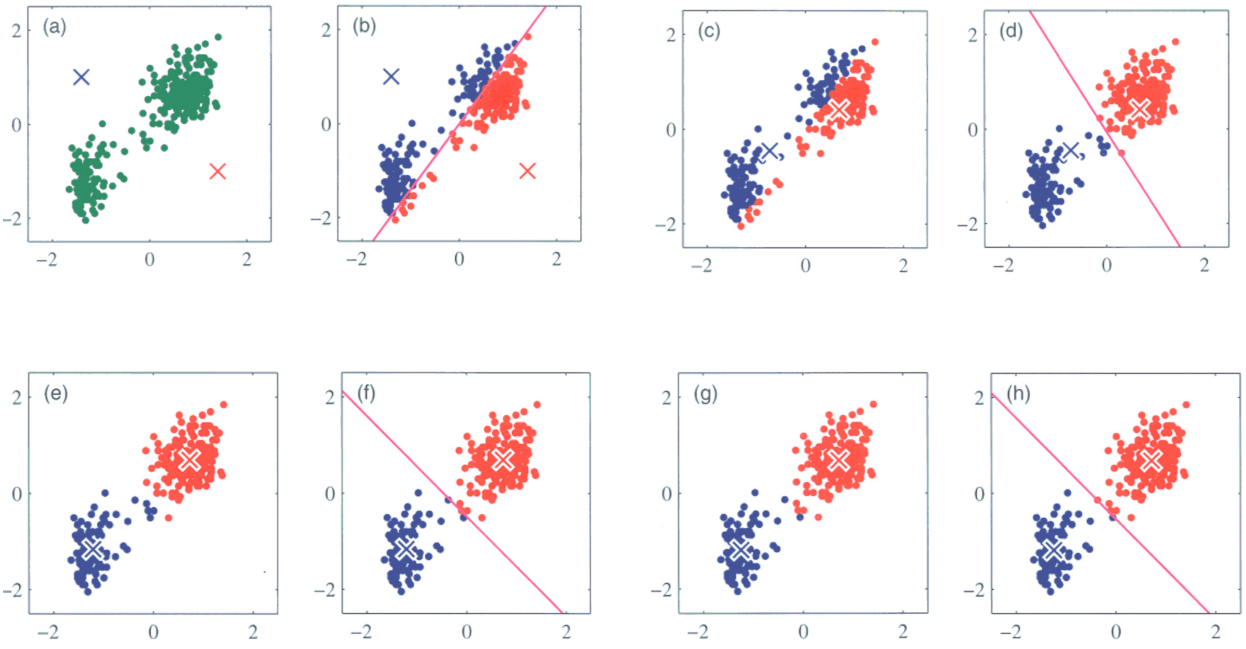
Algorithm

1. Randomly select K initial center.
2. Assign each data x to its closest center μ_i.
3. If the partition doesn't change: **stop**
4. Else, update the centers, each μ_i must be the gravity center of its cluster: $\mu_i = \frac{1}{|c_i|} \sum_{x \in c_i} x$
5. Go to 2

Like in most clustering algorithms, step 2 is dependent on a chosen distance function. For the K-Means algorithm, the Euclidian distance is usually preferred:

$$d(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

Example



Properties

Guaranteed to monotonically decrease average squared distance in each iteration

$$L(\mu)^{(t)} = \sum_{i=1}^{N} \min_j ||\mu_j - x_i||_2^2$$

$$L(\mu)^{(t+1)} \leq L(\mu)^{(t)}$$

- Convergence to a local minimum
- Algorithmic complexity at each iteration : O(n * d * K)
- Non-convex optimization: NP-hard problem

Limits

The K-Means algorithm si highly dependent no the initialization of the centers.

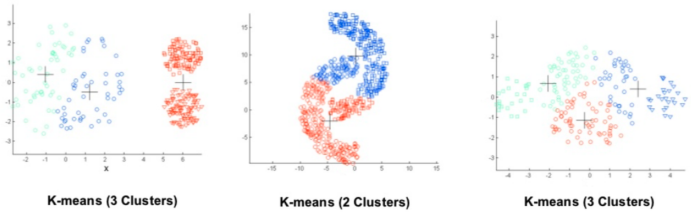
Instability

• This problem can be reduced by running the algorithm several time.
• It si also possible to initialize the centers using another clustering algorithm.

Clusters shape, size and density

• The K-Means algorithm can only capture *spherical* or almost spherical clusters. Any non-convex shape will be missed.

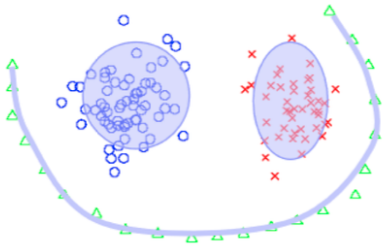
• Clusters with different density or sizes can also be difficult to find using the K-Means algorithm.



Number of clusters is a parameter: requires the number of clusters "K" to be provided as a parameter. In purely exploratory data mining tasks, this number is not always known and must be guessed.

Spectral Clustering

- Some data don't lend themselves for a centroid, a prototype-based, or a density-based clustering
- Spectral clustering proposes a different approach
- Spectral clustering treats the data clustering as a graph partitioning problem.



Example of a partitioning that most clustering methods cannot find

1. Let us note X= {x_1,.., x_n}, with $x_i \in R^d$ the data.
2. The points are characterized by pairwise similarities:

To each pair (x_i, x_j); we can associate a similarity value $w_{ij} = f(d(x_i, x_j), 0)$

1. The most common model is

$$w_{i,j} = \exp\left(\frac{-1}{2\sigma^2}||x_i - x_j||_2^2\right)$$

2. Let us note $W = (w_{ij})$ NxN this similarity matrix

Graph construction:

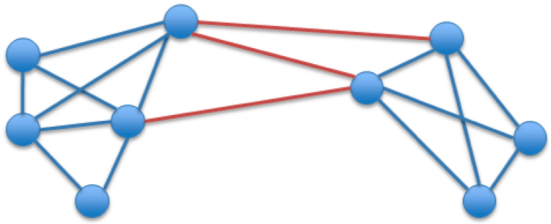
There are different ways to build a graph based on the similarity matrix:

- Fully connected graphs: All vertices having non-null similarities are connected to each other.
- Radius-based graphs: Each vertex is connected to all the vertices falling inside a ball of similarity of radius r.
- K-nearest neighbour graphs: Each vertex is connected to its K most similar neighbours.
- Hybrid KNN and radius-based graphs

In spectral clustering, the goal is to find two clusters that have the minimum weight sum connections:

This can be done by minimizing:

$$Cut(c_1, c_2) = \sum_{i \in c_1} \sum_{j \in c_2} w_{i,j}$$



It is easy to prove that the previous MinCut problem can be written as:

$$J_{MinCut} = \frac{1}{4} q^T (D - W) q$$

where D is a diagonal matrix so that $d_{ii} = \sum_{j=1}^{n} w_{ij}$, and q is a vector of size n so that:

$$q_i = \begin{cases} 1 & \text{if } x_i \in c_1 \\ -1 & \text{if } x_i \in c_2 \end{cases}$$

- We are therefore looking for the vector/partition q that minimizes this equation.
- This can be done by finding the eigenvalues and eigenvectors of the Laplacian matrix L= D- W.

Ideally, we also would like to maximize the intra-cluster similarity:

- First constraint: Minimize $Cut(c1, c2)$
- Second constraint: Maximize $Cut(c1, c1)$ and $Cut(c2, c2)$

This can be done by minimizing the following objective function:

$$J_{Ncut}(c_1, c_2) = Cut(c_1, c_2)\left(\frac{1}{\sum_{i \in c_1}\sum_{j=1}^{n} w_{ij}} + \frac{1}{\sum_{i \in c_2}\sum_{j=1}^{n} w_{ij}}\right)$$

–> It is proved that this minimization is obtained through the second smalest eigenvector of the matrix $L_{Ncut} = D^{-1/2}(D - W)D^{-1/2}$

STEPS:

Pre-processing

- Compute the similarity matrix and build the graph

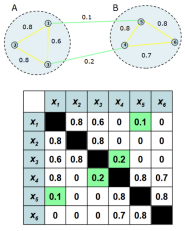
Spectral representation

- Compute the associated Laplacian Matrix.
- Compute the eigenvalues and eigenvectors.

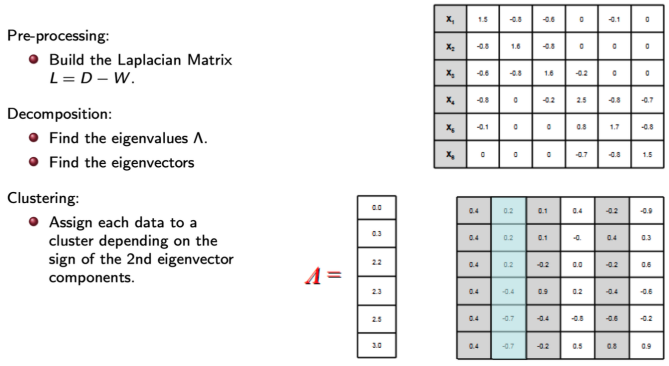
Clustering

- Assign each point to a cluster depending on the values in the second eigenvector.

EXAMPLE:



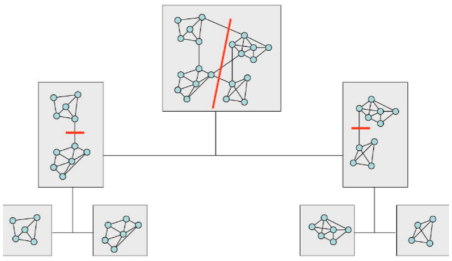
	x_1	x_2	x_3	x_4	x_5	x_6
x_1		0.8	0.6	0	0.1	0
x_2	0.8		0.8	0	0	0
x_3	0.6	0.8		0.2	0	0
x_4	0.8	0	0.2		0.8	0.7
x_5	0.1	0	0	0.8		0.8
x_6	0	0	0	0.7	0.8	



Extension to several clusters:

This method so far only allows to have 2 clusters. We would like to have more !

- Method 1: Recursively applying the algorithm in a hierarchical divisive manner.



- Method 2: Change the objective function to handle K clusters.

$$J_{Ncut}(c_1, \cdots, c_K) = \sum_{i=1}^{K} \frac{Cut(c_i, \bar{c}_i)}{\sum_{j \in c_i} \sum_{k=1}^{n} w_{jk}}$$

- Method 3: Use the other eigenvectors

  Combining the K-Means algorithm and spectral clustering

  - The matrix U containing eigenvectors 2 to K can be used as a low dimension representation of the data.
  - The matrix must first go through a unit normalization process:

$$Y_{ij} = \frac{U_{ij}}{\sqrt{\sum_{j=1}^{n} U_{ij}^2}}$$

  - After a unit normalization of each row, the K-Means algorithm (or any other clustering algorithm) can be applied to Y to obtain a partition with K clusters.

---

4. Open problems in clustering

Picking a similarity measure

Whatever the algorithm, they all rely on a similarity measure.

Picking a good similarity measure is therefore of a paramount importance.

Validating Clustering Results

Validating clustering results is a difficult process:

- There is no proper definition of what a "good cluster" is, or looks like.
- There is no "ground truth" in unsupervised learning to check the quality of a partition.
- Clusters come in all shapes and forms.
- The number of clusters to be found is usually unknown, and sometimes there are none.
- Not all data sets have well-defined clusters.

Internal Indexes

Internal indexes are criteria that can be used to evaluate the quality of a clustering partition.

- Internal indexes can be subjective as they favor some cluster shapes, and can be biased toward a lower number of clusters.
- They usualy assess the compactness of the clusters and whether they are wel separated.

Davies-Bouldin Index

It measures the average similarity between clusters and is based on the notion that good clusters should have low intra-cluster similarity and high inter-cluster dissimilarity.

Let S_i be the average scatter of a cluster ci around its mean value M_i:

$$S_i = \frac{1}{|c_i|} \sum_{x \in c_i} ||x - \mu||_2$$

The formula for calculating the Davies-Bouldin Index for a clustering partition with 'K' clusters is as follows:

$$DB = (1/K) * \sum_{i=1}^{K} max \frac{Si + Sj}{M_{i,j}}$$

Properties:

- A **lower** DB value means a **better** clustering.
- It favors spherical clusters.
- it is biased so that it gives lower values with less clusters

The Davies-Bouldin index is usualy the favored internal index used to evaluate partitions created using the K-Means algorithm.

Silhouette Index

The Silhouette index is another internal index with interesting properties

- It is normalized between 1- and 1 (a value bellow 0 meaning a bad partition).
- It can evaluate if a a data belongs to a cluster, if a cluster is well formed, or the whole partition.

Let $a_x \approx ||x - \mu||_2$ be the mean distance between $x \in c_i$ and the data that belong to the same cluster. And b_x the mean distance to the data that belong to other clusters.

$$b_x = \frac{1}{N - |c_i|} \sum_{y \notin c_i} ||x - y||_2 = \sum_{k \neq i} \frac{|c_k|}{N - |c_i|} ||x - \mu_k||_2$$

Silhouette index of an element x

$$SC(x \in c_i) = \frac{b_x - a_x}{max(a_x, b_x)}$$

Silhouette index of a cluster c_i

$$SC(c_i) = \frac{1}{|c_i|} \sum_{x \in c_i} SC(x)$$

Silhouette index of a partition

$$SC = \frac{1}{K} \sum_{i=1}^{K} SC(c_i)$$

- The silhouette index best value is 1.
- It favors spherical clusters.

Range	Interpretation
$S > 0.70$	Strong structures have been found
$0.5 < S < 0.70$	reasonable structures have been found
$0.25 < S < 0.5$	Weak structures have been found
$0 < S < 0.25$	No substantial structure found
$S < 0$	This clustering is most likely garbage

Table: Interpreting Silhouette values

C-Index

This index is defined as follows:

$$C = \frac{S - S_{min}}{S_{max} - S_{min}}$$

- S is the sum of distances over all pairs within the same clusters.
- Let there be I of these pairs.
- S_min is the sum of the / smalest distances
- S_max is the sum of the / largest distances

Calinski & Harabasz

$$CH(k) = \frac{B/(k-1)}{W/(N-k)}$$

- B is the sum of squares among the clusters
- W is the sum of squares within clusters
- k is the number of clusters

The notion of stability

A cluster, or a partition are said to be **stable** if they are **insensitive** to **small changes** in the underlying data.

- Stability is usually tested by bootstrapping the same clustering algorithm many times over several sub-samples of the data.

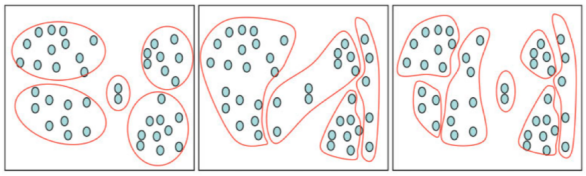


Figure: Example of an unstable partition

Given the results of R simulations C1, ,.•Cr, we have:

$$S = \frac{1}{(R-1)^2} \sum_{i=1}^{R} \sum_{j \neq i}^{R} (1 - \Delta(C_i, C_j))$$

External indexes

Quite often clustering algorithms wil be tested ni a non-exploratory setting:

- A data set for which the real classes are known will be used for test purposes (with the added advantage that the number of clusters will be known)
- In this case, the cluster found will be compared to the real classes using external indexes or purity measures.

Cluster purity

Given K real classes $Y^1, \cdots, Y^K$:

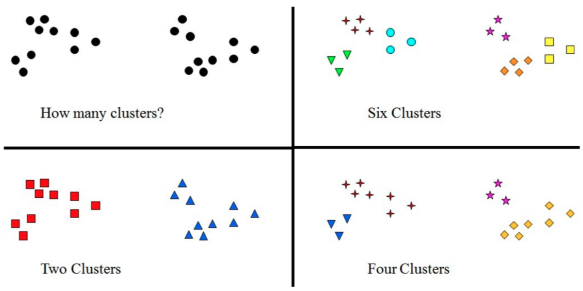
- Purity of a cluster C^i: $CP(C^i) = \frac{1}{|C^i|} argmax_j |C^i \cap Y^j|$
- Purity of a partition: $CP = \frac{1}{N} \sum_{i=1}^{k} argmax_j |C^i \cap Y^j|$

Examples of external indexes

- Rand Index, Adjusted Rand Index, Accuracy, etc.

How many clusters?

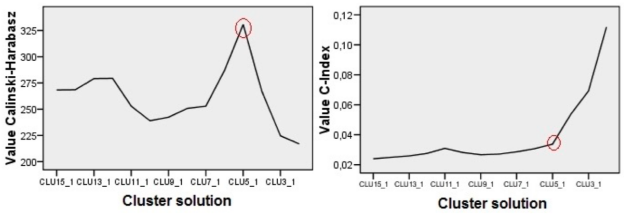
Guessing the right number of clusters is one of the oldest problem in clustering, and several methods require it as a parameter.

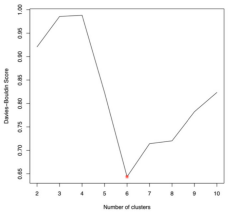


Some criteria exist to pick the optimal number of clusters, but they require to run the algorithm several times with different number of clusters in order to evaluate the solutions:

- The Bayes Information Criterion (BIC) and Akaike Information Criterion (AIC) work well with probabilistic algorithms.
- The Minimum Description Length criterion (MDL) and the Minimum Message Length criterion (MML) are recommended for hierarchical clustering.
- Stability can also be used to guess the right number of clusters.

Sometimes clustering indexes can be used to help picking the right number of clusters:





Notes3

1. Introduction

Visualization: Is a branch of computer science involving the processing, analysis and graphical representation of data from diverse fields. One goeal is to visually represent data that does not necessarily have a natural geometric interpretation.

Plot Unvariate Data

Histogram, Pie Plot, Box Plot.

Plot Bivariate Data

Biplot, Time series trend

Curse of dimensionality

The "curse of dimensionality" is a term introduced in 1961 by Bellman refering to the problem of the **explosive increase** in **data volume** associated with adding **extra dimensions** in a mathematical space.

- High dimensional data are difficult to efficiently visualize in 2D without loosing meaningful information.
- While computer algorithms can grasp information in higher dimensional spaces than humans, past some point, they too have trouble extracting the meaningful information.

We will illustrate this problem with a simple example.

Toy Problem

We consider a Pattern Recognition problem with 3 classes. We have 9 available observation represented in 1 dimension

A simple approach would be to do the following:

- Divide the feature space into uniform bins.
- Compute the class ratio in each bin and use a majority vote to classify the bin
- When a new example comes, put it in its closest bin

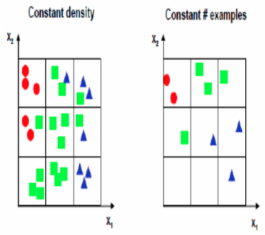
I this first example with one feature, we observe that when dividing the space into 3 segments we get 3 homogeneous regions each with a similar density of 3 examples per bin

We also see that there is too much overlap between the classes, so we decide to add a second feature to try to improve the class separability.

If we add a 2nd dimension we pass from 3 cases in 1D to 3*3 = 9 in 2D.

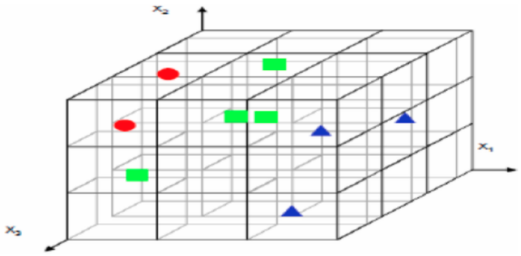
We have an another problem: do we maintain the density of examples per bin or do we keep the same number of example than in 1D?

- If we want to maintain the density, we now ned 72 examples instead of 9.
- Choosing to maintain the number of examples to 9 will result in a very sparse 2D scatter plot.



Adding a 3rd dimension makes the problem worse:

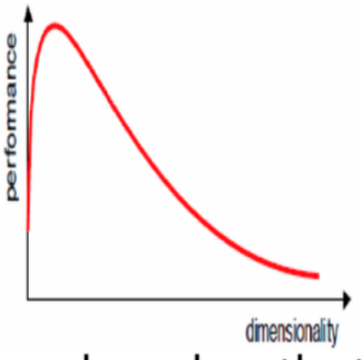
- We move to 3 3 = 72 bins
- Keeping aconstant density now requires 18 observations.
- Keeping the same number of observation gives us an almost empty 3D plot.



The curse of dimensionality generates several phenomenons such as:

- The concentration of observations in given space areas
- The desertification of the data space
- The depopulation of the center of hyper-volumes

I practice, the curse of dimensionality means that, for a given sample size, there is a maximum number of variables beyond which a classifier performance will degrade rather than improve.



How to deal with the curse of dimensionality?

These findings suggest that we need special treatment to manipulate high dimensional data:

- Incorporating prior knowledge to reduce the search space.
- Providing smoother and more generic target functions.
- Reducing dimensionality.

Reducing dimensionality

The reduction of dimensionality by the mean of space transformations, projections, or feature selection is usually the prefered solution in combination with the two others.

- Prior knowledge is not always available, and may alone result in over-fitting.
- Smoother target functions alone may result in a global decrease of performances and are cumbersome to design.

2. Dimension reduction via feature *selection*

Dimension Reduction

Dimension reduction in based on the hypotesis that high dimensional data are not uniformly distributed → Dimension reduction looks for structures that define there high and low density areas to tackle the curse of dimensionality

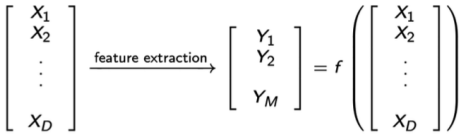
There are two main methodology for dimension reduction:

- Feature selection:

Choosing a subset of al the features.

- Feature extraction:

Creating a small set of new features by combinations of the original ones.



Feature Selection

Definition: Feature selection is a process to choose an optimal subset of relevant variables from a set of variables, according to a performance criterion

Question 1: How to measure the relevance of the variables?

> We need to find a measure of relevance or evaluating criterion J(X) to quantify the importance of one variable or a combination of variables

Question 2: How to obtain the optimal subset?

> To answer this question we need to define a procedure or algorithm that will determine the optimal subset of relevant variables based on the evaluation criterion

Question 3: Which optimal criterion to use?

> With the answer to Q2 begin an optimization algorithm, we need to define a stopping citerion for this optimization process.

Examples for choosing a criterion…

Branch & Bound based methods

Branch & Bound based methods use an optimization strategy that finds the optimal subset of variable based on a minimal bound to reach on the evaluation criterion.

Algorithm:

1. The algorithm starts with all features and removes one at each step.
2. If the evaluation criterion scores bellow the bound, the algorithm backtracks and tries to remove another feature.
3. Groups of features bellow the bound are remembered so that sub-sets of these groups are never tested later in the process.
4. When there is no feature left to try removing, the algorithm remembers the best score and backtracks to another branch.
5. The algorithm stops when all branches with valid subsets have been tested.

Branch & Bound methods can only work if the selection criterion J(X) is monotonic

Monotonic criterion: Definition

Let us consider two subsets of features A_1 and A_2. Then, $J(\cdot)$ is monotonic if and only if $A_1 \subset A_2 \implies J(A_1) \leq J(A_2)$.

Problem: Most effective evaluation criteria are **not** monotonic

The use of sub-optimal methods si then required:

1. Sequential Forward Selection (SFS)
2. Sequential Backward Selection (SBS)
3. Bidirectional Selection (BS)

1. Sequential Forward Selection

Let $X = \{X_1, \cdots, X_D\}$ be a set of variables. The SFS procedure is the following:

- Initially set the selected set of variables A_0 as empty.
- At each step k, select the variable X_i that maximizes the following criterion of evaluation $J(A_k)$:

$$J(A_k) = \max_{X_i \in (X \setminus A_{k-1})} J(A_{k-1} \cup X_i)$$

This will result in a list of variables ordered by their importance. The procedure can be stopped at any step when a given criterion (e.g. acceptable loss of information) has reached an acceptable value.

2. Sequential Backward Selection

Let $X = \{X_1, \cdots, X_D\}$ be a set of variables. The SBS procedure is the following:

- Initially start with a set $A_D = X$ containing all the variables.
- At each step for $k \in [D-1 \cdots 1]$, remove the variable X_i of the least importance according to:

$$J(A_k) = \max_{X_i \in A_{k+1}} J(A_{k+1} \setminus X_i)$$

This will also result in a list of variables ordered by their importance, with the most important being at the end of the list. The procedure can also be stopped has soon as the remaining set is bellow an acceptable lower bound of accuracy on a given criterion.

3. Bidirectional Selection

The Bidirectional Selection procedure performs the search in both Forward and Backward direction in a competitive manner:

The procedure stops when one of the following case occurs:

- One of the two direction has found the best subset of variables before reaching the middle of the search space.
- The two branches are reaching the middle.

This method reduces the search time as the search is performed in both directions and stops when there is a solution regardless of the direction.

Remark

The sets of best selected variables found respectively by SFS and SBS are not equal and rarely identical because of their different principles of selection.

3. Dimension reduction via feature *extraction*

Feature extraction consists in building new features from the original ones with one or several of the following goals:

- Having a lower number of features while keeping a maximum of information
- Having better features with which the data are easier to process
- Having feature with which the data are easier ot visualize

There are several methods for feature extraction:

Linear Methods: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA)

Non Lineare Methods: Isometric feature mapping (Isomap), Localy Linear Embedding (LLE)

PCA

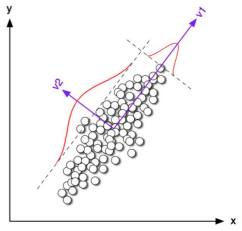
PCA is well known feature extraction algorithm that we have already seen.

It builds a low number of new features based on linear combinations of the original ones.

Algorithm:

PCA Algorithm

1. Choose p the number of dimensions that you want to keep.
2. Select the eigenvectors associated with the p largest eigenvalues of the variance-covariance matrix.
3. Project the data on the new axes in order to obtain a description of these data according to the new variables (called principal components).
4. If the number of dimensions is ≤ 3, it is possible to represent the data in the new space.



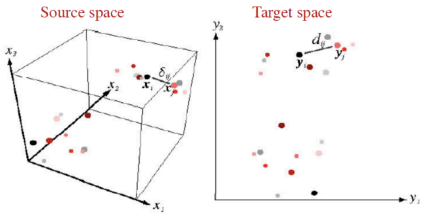
Changing the axis using PCA for more convenient variables

PCA being based on the variance-covariance matrix, it can be used to draw correlation circles. In turn, the projection of data on correlation circles can be useful for visualization purposes.

Multidimensional Scaling

Multi dimensional scaliing MDS:

- MDS encompasses a collection of reduction techniques that maps the distances between observations in a high dimensional space into a lower dimensional one.
- It aims at finding a configurations of the points in the low dimensional space so that the inter-point distances remain mostly proportional to the one in higher dimension



Besides dimension reduction< MDS in well suited for applications or algorithms based on the distances between the data, rather than the data themselves:

- Rebuilding the map of country based on the distance between the cities.
- Building 2d or 3d map of the coordinates

Remarks

- With $\frac{N(N-1)}{2}$ distances, it is always possible to generate the position of N points in a N dimension space.
- MDS computes an approximation for a lower dimensional space.

Basically:

We have

- The points $x_1, \cdots, x_N$ in D dimensions
- Any distance δ_{ij} between points x_i and x_j

We want to find

- The points $y_1, \cdots, y_N$ in 2 or 3 dimensions
- The distance d_{ij} between any y_i and y_j so that it is close to δ_{ij}

Cost function:

We must search fro d_{ij} s that it minimizes on abjective function.

We can define such in a general manner (there are several of them):

$$Cost = \sum_{i<j} (d_{ij} - \delta_{ij})^2$$

$$\delta_{ij} = ||x_i - x_j||^2$$

$$d_{ij} = ||y_i - y_j||^2$$

Algorithm:

The MDS algorithm involves the following steps:

- Choose an objective function J and a learning rate
- Compute the distances delta_{ij} if they are not provided
- Initialize the points y1,…,yN randomly and compute the corresponding d_{ij}
- Repeat until convergence

$$\forall i \quad y_i \leftarrow y_i - \eta \nabla J(y_i)$$

Example 1:

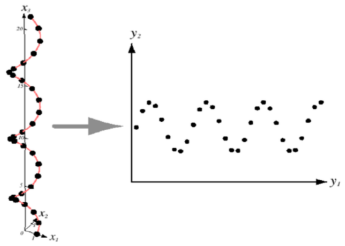


Figure: From 3D to 2D using MDS

Example 2:

Eurdist, the dataset containing the distances in kilometers between 21 European cities

The source set is a square matrix containing all the distances between the cities.

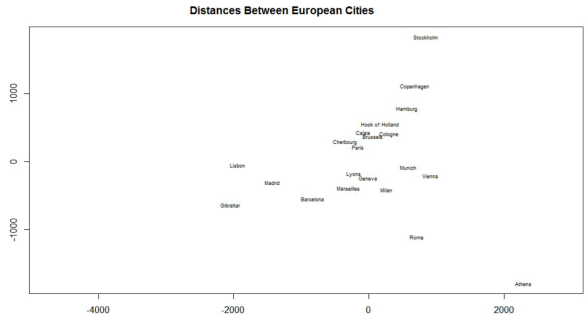


Figure: Reconstructed map of Europe using MDS

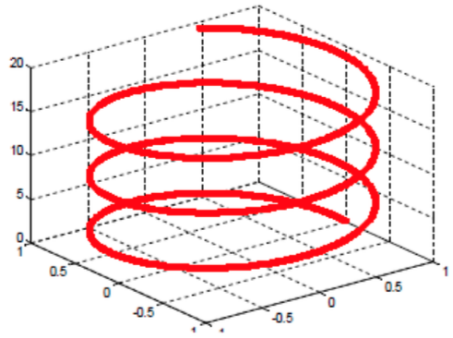
Limits of linear methods

Techniques such as LDA, PCA and their variants perform a global transformation of the data using basic operations:

- Rotation
- Translation
- Rescaling

- Using the 3 operations, linear methods **assume** that most of the information in the data is contained in a linear subspace.
- This raises the question of how to deal with data that are actually embedded in a non-linear subspace a (low-dimensional manifold).

Linear methods such as PCA cannot discover the structure of a data set with a non-linear shape.



- **Euclidian based distances** are one of the main problem when dealing with data embedded in a non-linear subspace.
- Custom distances embedded in the low-dimensional manifold, such as the **geodesic distance**, can help dealing with such data.

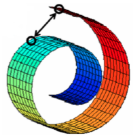


Figure: Euclidian distance

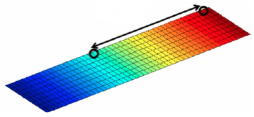


Figure: Geodesic distance

ISOMAPP

Isometric feature mapping is a reduction method that can handle non-linear subspaces. It is based on the following principles:

- For neighbouring samples (close data points) the euclidian distance provides a good approximation of the geodesic distances
- For distant points, the geodesic distance can be approximated using an Euclidian distance-based neighbourhood graph of the data and searching the shortest path between the 2 points

Algorithm:

1. Build a distance neighborhood graph G using the Euclidian distance in the input space: This can be done using KNN, or by connecting the points within a radius ϵ.
2. Approximate the geodesic distance between all data points: Compute the shortest path between all points of the graph using **Dijkstra** algorithm or **Floyd** algorithm.
3. Apply the **Multi-dimensional scaling** (MDS) algorithm to the geodesic distance matrix from Step 2.

Remark

The Isomap algorithm can be seen as a non-linear variant of the MDS algorithm.

Algorithmic complexity:

The main issue of the Isomap algorithm is that it can be quite slow. For a data set of size N, an input space of dimension D, an output space of dimension d, and considering a neighborhood of size k ni the first step, we have:

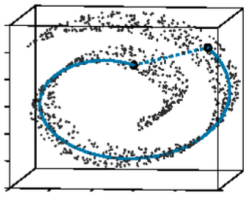
- Step 1: K-nearest neighbors O(N^D)
- Step 2: Djikstra O(N^2 log(N) +N^2 k), Floyd O(N^3)
- Step 3: MDS algorithm O(N^d)

Example:

The "swiss roll" data set contains 20000 points.

We will show thereafter the development of the Isomap algorithm on this data set.

A 1000 points extract of the Swiss roll data set,
with an example of Euclidian and Geodesic distances.



In the figure bellow, we show 3 illustrations of how the Isomap algorithm works on the Swiss roll data set with the following parameters:

- K-nearest neighbors (K=7)
- Approximation of the Geodesic distance: Djikstra algorithm

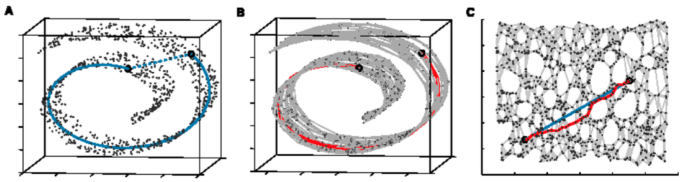


Figure: A 1000 points extract of the Swiss roll data set: The Euclidian distance is in dotted blue, the Geodesic distance in solid blue, and the approximated Geodesic distance is in red. [Tenebaum et al. 2000]

Advantages:

Handles non-linearity

Is non iterative

Preserve the global properties of the data

Limitations:

Sensitive to noise

The optimal parameters k for KNN or e can be difiicult to determine

Relatively slow with large data sets

4.Dimension reduction using unsupervised learning

Spectral Clustering

Reminder:

1. Can detect clusters with all sorts of shapes
2. It's relatively fast with sparse data (often the case in high dimension)

Those 2 properties show that spectral clustering may have hidden properties that are useful for dimension reduction.

Algorithm:

1. Compute the similarity matrix W
2. Build the similarity graph using KNN, or connecting neighbors based on a radius ϵ
3. Compute the Laplacian matrix $L = D - W$, where D is diagonal with $d_{ii} = \sum_{j=1}^{N} w_{ij}$
4. Compute the eigenvalues and eigenvectors associated to L.
5. Do either:
   - Split into 2 clusters using the second eigenvector
   - Use eigenvectors 2 to d as a low dimension representation of the data and apply another clustering algorithm on it.

Remark

It very much looks like the steps of the non-linear dimension reduction methods that we have just seen.

To use Sp. Cl. for dimensional reduction purposes, the eigenvectors need to be transformed in order to keep good topological properties. The clustering step in then remove as it is not needed here.

Transforming the eigenvectors into good low dimensional representations

- Let us define U the matrix containing eigenvectors 2 to d can be used to represent the data in a d-dimensional space.
- The matrix must first go through a unit normalization process:

$$Y_{ij} = \frac{U_{ij}}{\sqrt{\sum_{j=1}^{n} U_{ij}^2}}$$