Data Analysis

Mariem Gzara

Master professionnel en Génie Logiciel

Institut Supèrieur d'Informatique et de Mathématique de Monastir

Data Analysis

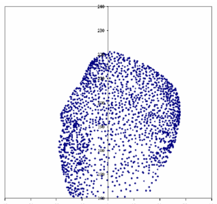
Introduction

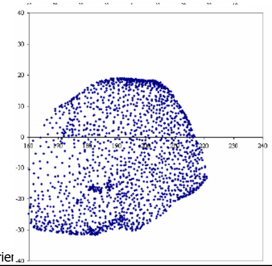
- Why Data analysis?
  - Given this data set of points in the 3-dimensional space, what type of information we are expected to leran from this data?

IND	X	Y	Z
1	52,58	-25,13	-101,93
2	51,5	-24,86	-101,77
3	52,49	-23,58	-101,61
4	52,03	-26,22	-101,55
5	54,32	-23,64	-101,44
6	53,63	-26,4	-101,4
7	50,47	-25,56	-101,36
...	...	...	...
21488	-49,21	79,23	60,14
21489	-47,28	86,02	60,15
21490	-46,34	87,09	60,16
21491	-48,21	81,83	60,19
21492	-46,91	79,37	60,19
21493	-43,92	87,68	60,2
21494	-47,29	84,45	60,23
21495	-45,55	81,08	60,23
21496	-44,41	83,48	60,25
21497	-43,46	85,53	60,26
21498	-45,39	85,67	60,3
21499	-45,39	85,67	60,3

Introduction

- And if I project the data points on the plan X-Y or the plan Y-Z, what information can i learn from it?
- The rate of data creation is accelerating each year
- More the amount of the data are
- As the amount of data increases, the proportion of data increases.
- How to turn large amount of data into information … knowledge …. profit





Data Analysis

Chapter 1
Data Exploration and Preparation

Chapter 1: Data Exploration and Preparation

- General data characteristics
- Basic data description and exploration
- Data cleaning
- Data integration and transformation
- Data reduction

Types of Data Sets

- Record
  - Relational records
  - Data matrix, e.g., numerical matrix, crosstabs
  - Document data: text documents: term-frequency vector
  - Transaction data
- Graph
  - World Wide Web
  - Social or information networks
  - Molecular Structures
- Ordered
  - Spatial data: maps
  - Temporal data: time-series
  - Sequential Data: transaction sequences
  - Genetic sequence data

	team	coach	pla y	ball	score	game	wi n	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

TID	Items
1	Bread, Coke, Milk
2	Juice , Bread
3	Juice, Coke, Diaper, Milk
4	Juice, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Important Characteristics of Structured Data

- Dimensionality
  - Curse of dimensionality
- Sparsity
  - Only presence counts
- Resolution
  - Patterns depend on the scale
- Similarity
  - Distance measure

Types of Attribute Values

- Nominal
  - E.g., profession, ID numbers, eye color, zip codes
- Ordinal
  - E.g., rankings (e.g., army, professions), grades, height in {tall, medium, short}
- Binary
  - E.g., medical test (positive vs. negative)
- Interval
  - E.g., calendar dates, body temperatures
- Ratio
  - E.g., temperature in Kelvin, length, time, counts
- Discrete :
  - zip codes, profession, key words
- Continuous Attribute: typically represented as floating point variables
  - temperature, height, or weight

Chapter 1: Data Exploration and Preparation

- General data characteristics
- Basic data description and exploration
- Data cleaning
- Data integration and transformation
- Data reduction
- Summary

Measuring the Central Tendency

- Mean (algebraic measure) (sample vs. population):
  - Mean

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

  - Weighted arithmetic mean:

$$\overline{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

  - Trimmed mean: chopping extreme values

$$\mu = \frac{\sum x}{N}$$

Measuring the Central Tendency

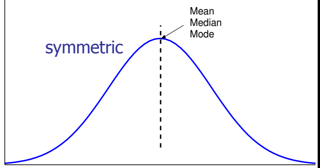
- Median:
  - Middle value if odd number of values, or average of the middle two values otherwise
  - Estimated by interpolation (for *grouped data*):

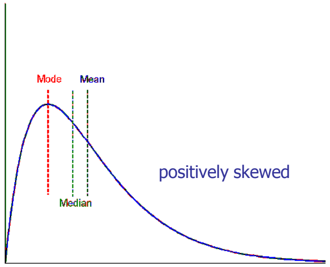
$$median = L_1 + \left(\frac{N/2 - (\sum freq)l}{freq_{median}}\right) width$$

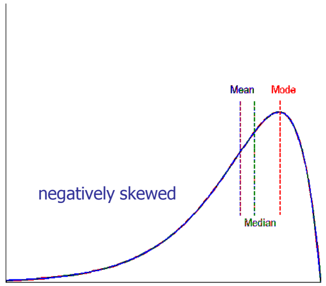
- Mode
  - Value that occurs most frequently in the data
  - Unimodal, bimodal, trimodal
  - Empirical formula: $mean - mode = 3 \times (mean - median)$

Symmetric vs. Skewed Data

- Median, mean and mode of symmetric, positively and negatively skewed data







Measuring the Dispersion of Data

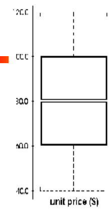
- Variance and standard deviation (*sample: s, population: σ*)
  - Variance: (algebraic, scalable computation)
  - Standard deviation *s (or σ)* is the square root of variance *s^2 (or σ^2)*

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n}[\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2]$$

Measuring the Dispersion of Data

- Quartiles, outliers and boxplots
  - Quartiles: Q_1 (25^{th} percentile), Q_3 (75^{th} percentile)
  - Inter-quartile range: IQR = $Q_3 - Q_1$
  - Five number summary: min, Q_1, M, Q_3, max

Boxplot Analysis

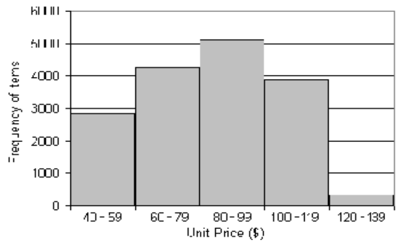


- Five-number summary of a distribution:

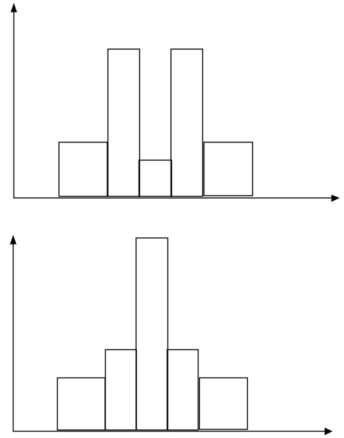
  Minimum, Q1, M, Q3, Maximum
- Boxplot
  - Data is represented with a box
  - The ends of the box are at the first and third quartiles, i.e., the height of the box is IQR
  - The median is marked by a line within the box
  - Whiskers: two lines outside the box extend to Minimum and Maximum
  - Outlier: usually, a value higher/lower than 1.5 x IQR

Histogram Analysis

- Histogram: x-axis are values, y-axis repres. frequencies
- Graph displays of basic statistical class descriptions
  - Frequency histograms
    - A univariate graphical method
    - Consists of a set of rectangles that reflect the counts or frequencies of the classes present in the given data



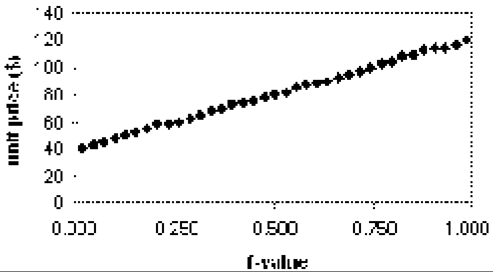
Histograms Often Tells More than Boxplots



- The two histograms shown in the left may have the same boxplot representation
  - The same values for: min, Q1, median, Q3, max
- But they have rather different data distributions

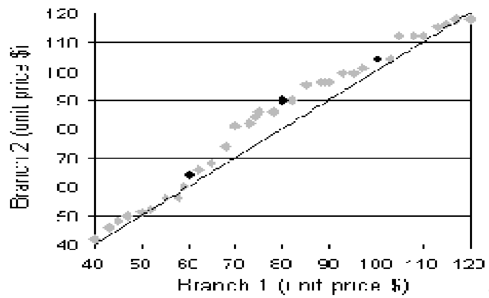
Quantile Plot

- Displays all of the data (allowing the user to assess both the overall behavior and unusual occurrences)
- Plots quantile information
  - For a data x_i data sorted in increasing order, f_i indicates that approximately 100 f_i% of the data are below or equal to the value x_i



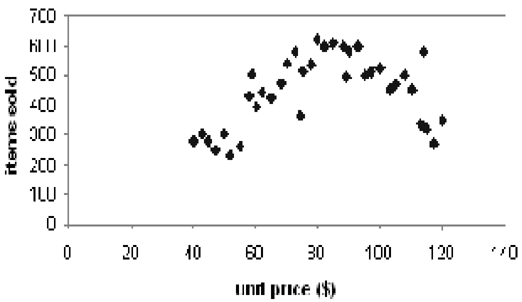
Quantile-Quantile (Q-Q) Plot

- Quantile-quantile (q-q) plot: graphs the quantiles of one univariant distribution against the corresponding quantiles of another
- Graphs the quantiles of one univariate distribution against the corresponding quantiles of another
- Allows user to view whether there is a shift in going from one distribution to another



Scatter plot

- Provides a first look at bivariate data to see clusters of points, outliers, etc
- Each pair of values is treated as a pair of coordinates and plotted as points in the plane



Chapter1: Data Exploration and Preparation

- General data characteristics
- Basic data description and exploration
- Data cleaning
- Data integration and transformation
- Data reduction
- Summary

Why Is Data Dirty?

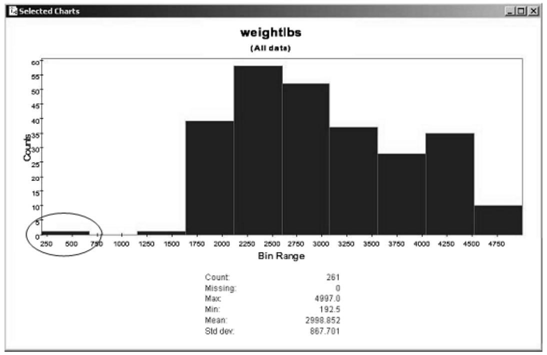
- Incomplete data may come from
  - "Not applicable" data value when collected
  - Different considerations between the time when the data was collected and when it is analyzed.
  - Human/hardware/software problems
- Noisy data (incorrect values) may come from
  - Faulty data collection instruments
  - Human or computer error at data entry
  - Errors in data transmission
- Inconsistent data may come from
  - Different data sources
  - Functional dependency violation (e.g., modify some linked data)
- Duplicate records also need data cleaning

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
  - a global constant : e.g., "unknown", a new class?!
  - the attribute mean
  - the attribute mean for all samples belonging to the same class: smarter
  - the most probable value: inference-based such as Bayesian formula or decision tree

How to identify outliers ?

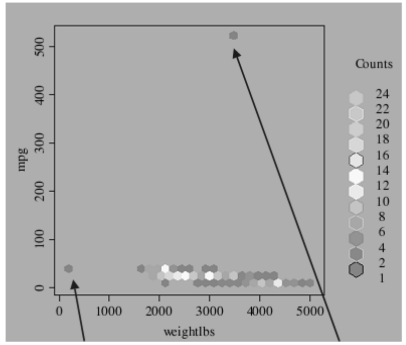
- Graphical methods for identifying outliers



Histogram of vehicle weights: can you find the outlier?

How to identify outliers ?

- Graphical methods for identifying outliers



Scatter plot of *mpg* against *weightlbs* shows two outliers.

- A data value is an outlier if:

a. It is located 1.5(IQR) or m

b. It is located 1.5(IQR) or m

How to Handle Noisy Data?

- Binning
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
  - smooth by fitting the data into regression functions
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

Simple Discretization Methods: Binning

- Equal-width (distance) partitioning
  - Divides the range into N intervals of equal size: uniform grid
  - if A and B are the lowest and highest values of the attribute, the width of intervals will be: $W = (B-A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into N intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky

Binning Methods for Data Smoothing

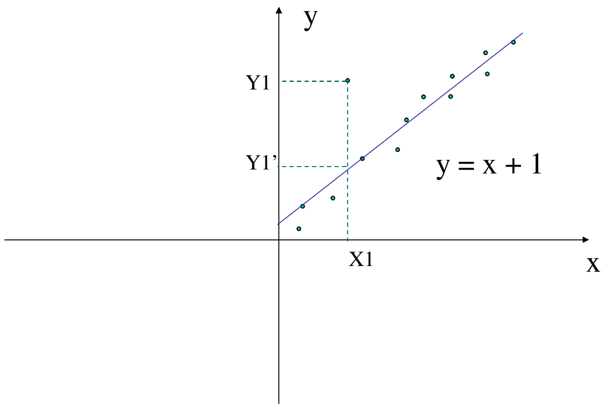
- Sorted data for price : 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Partition into equal-frequency (equi-depth) bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34

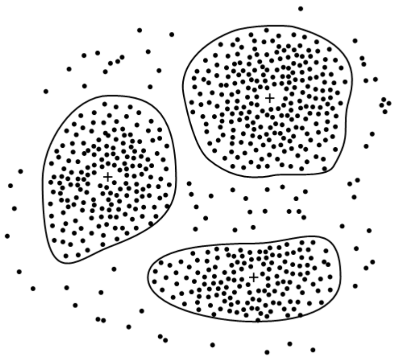
* Smoothing by bin means:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29

* Smoothing by bin boundaries:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

Regression



Cluster Analysis



A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters. Each cluster "center" is marked with a "+".

Chapter 1: Data Exploration and Preparation

- General data characteristics
- Basic data description and exploration
- Measuring data similarity
- Data cleaning
- Data transformation
- Data reduction
- Summary

Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values s.t. each old value can be identified with one of the new values
- Methods
  - Smoothing: Remove noise from data
  - Normalization: Scaled to fall within a small, specified range
    - min-max normalization
    - z-score normalization
    - normalization by decimal scaling
  - Attribute/feature construction
    - New attributes constructed from the given ones

Data Transformation: Normalization

- Min-max normalization: to [new_min_A, new_max_A]

$$v' = \frac{v - min_A}{max_A - min_A}$$

  - Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,000 is mapped to $\frac{73,600-12,000}{98,000-12,000} = 0.716$
- Z-score normalization (μ: mean, σ: standard deviation):

$$v' = \frac{v - mean_A}{std_A}$$

  - Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73,600-54,000}{16,000} = 1.225$
- Normalization by decimal scaling

$$v' = \frac{v}{10^j}$$ Where j is the smallest integer such that Max(|v'|) < 1

Chapter 1: Data Exploration and Preparation

- General data characteristics
- Basic data description and exploration
- Measuring data similarity
- Data cleaning
- Data transformation
- Data reduction

Data Reduction Strategies

- Why data reduction?
  - A database/data warehouse may store terabytes of data
  - Complex data analysis/mining may take a very long time to run on the complete data set
- Data reduction: Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results
  - Dimensionality reduction — e.g., remove unimportant attributes
    - Principal component analysis
    - Singular value decomposition
    - Supervised and nonlinear techniques (e.g., feature selection)
  - Numerosity reduction (some simply call it: Data Reduction)
    - Data cub aggregation
    - Data compression
    - Regression
    - Discretization (and concept hierarchy generation)

Data analysis
Cluster analysis
— Chapter 2—

Mariem Gzara

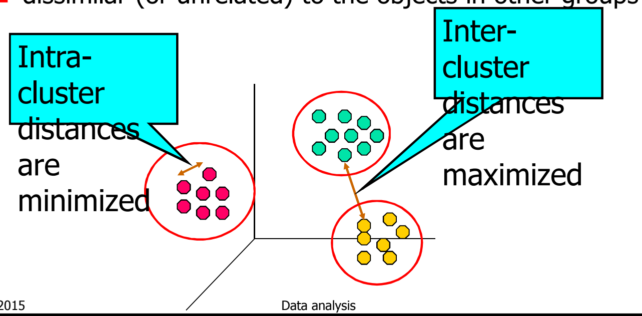
Master professionnel Génie Logiciel

Institut Supèrieur d'Informatique et de Mathématique de Monastir

Chapter 2. Cluster Analysis

What is Cluster Analysis?

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups



Clustering for Data Understanding and Applications

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market research

Clustering as Preprocessing Tools (Utility)

- Summarization:
  - Preprocessing for regression, classification, and association analysis
- Compression:
  - Image processing: vector quantization
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters

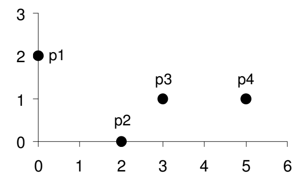
Chapter 2. Cluster Analysis

1. What is Cluster Analysis?
2. Similarity/ dissimilarity
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Graph-Based Methods

Similarity and Dissimilarity

- Similarity
  - Numerical measure of how alike two data objects are
  - Value is higher when objects are more alike
  - Often falls in the range [0,1]
- Dissimilarity (i.e., distance)
  - Numerical measure of how different are two data objects
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- Proximity refers to a similarity or dissimilarity

Example: Data Matrix and Distance Matrix



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Data Matrix

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix (i.e., Dissimilarity Matrix) for Euclidean Distance

$$d(pi,pj)=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$$

Minkowski Distance

- Properties
  - d(i, j) > 0 if i ≠ j, and d(i, i) = 0 (Positive definiteness)
  - d(i, j) = d(j, i) (Symmetry)
  - d(i, j) ≤ d(i, k) + d(k, j) (Triangle Inequality)
- A distance that satisfies these properties is a metric
- *Minkowski distance*: A popular distance measure

$$d(i,j) = \sqrt[k]{(|x_{i1}-x_{j1}|^k + |x_{i2}-x_{j2}|^k + \dots + |x_{ip}-x_{jp}|^k)}$$

where $i = (x_{i1}, x_{i2}, \dots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \dots, x_{jp})$ are two p-dimensional data objects, and k is the order

Special Cases of Minkowski Distance

- $k = 1$: Manhattan (city block, L_1 norm) distance
  - E.g., the Hamming distance: the number of bits that are different between two binary vectors

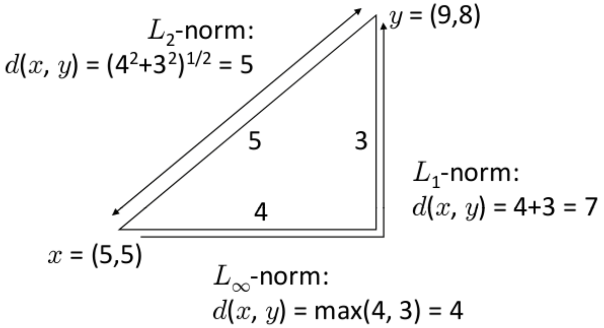
$$d(i,j) = |x_{i1}-x_{j1}| + |x_{i2}-x_{j2}| + \dots + |x_{ip}-x_{jp}|$$

- $k = 2$: (L_2 norm) Euclidean distance

$$d(i,j) = \sqrt{(|x_{i1}-x_{j1}|^2 + |x_{i2}-x_{j2}|^2 + \dots + |x_{ip}-x_{jp}|^2)}$$

- $k \to \infty$. "supremum" (L_{max} norm, L_∞ norm) distance.
  - This is the maximum difference between any component of the vectors
- Do not confuse k with n, i.e., all these distances are defined for all numbers of dimensions.

Special Cases of Minkowski Distance



Unit circles in 2d (source: Wikipedia):

Distances

- Canberra distance

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| (x_i + y_i)$$

- Maximum distance

$$d(x, y) = \max_i |x_i - y_i|$$

- Also, one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures

Example: Minkowski Distance

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

Interval-valued variables

- Standardize data
  - Calculate the mean absolute deviation:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + ... + |x_{nf} - m_f|)$$

where $m_f = \frac{1}{n}(x_{1f} + x_{2f} + ... + x_{nf})$.

  - Calculate the standardized measurement (*z-score*)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- Using mean absolute deviation is more robust than using standard deviation
- Then calculate the Enclidean distance of other Minkowski distance

Binary Variables

- A contingency table for binary data

		Object j		
		1	0	*sum*
Object i	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

- Distance measure for symmetric binary variables:

$$d(i, j) = \frac{b + c}{a + b + c + d}$$

- Distance measure for asymmetric binary variables:

$$d(i, j) = \frac{b + c}{a + b + c}$$

Binary Variables

- Jaccard coefficient (similarity measure for asymmetric binary variables):

$$sim_{Jaccard}(i, j) = \frac{a}{a+b+c}$$

- Note: Jaccard coefficient is the same as "coherence":

$$coherence\ (i, j) = \frac{\sup(i, j)}{\sup(i) + \sup(j) - \sup(i, j)} = \frac{a}{(a+b)+(a+c)-a}$$

Dissimilarity between Binary Variables

- Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

  - gender is a symmetric attribute
  - the remaining attributes are asymmetric binary
  - let the values Y and P be set to 1, and the value N be set to 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

$$d(i, j) = \frac{p - m}{p}$$

- Method 2: Use a large number of binary variables
  - creating a new binary variable for each of the *M* nominal states

Ordinal Variables

- An ordinal variable can be discrete or continuous
- Order is important, e.g., rank
- Can be treated like interval-scaled
  - replace x_{if} by their rank $r_{if} \in \{1, ..., M_f\}$
  - map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  - compute the dissimilarity using methods for interval-scaled variables

Ratio-Scaled Variables

- Ratio-scaled variable: a positive measurement on a nonlinear scale, approximately at exponential scale, such as Ae^{Bt} or Ae^{-Bt}
- Methods:
  - treat them like interval-scaled variables—*not a good choice!* (why?—the scale can be distorted)
  - apply logarithmic transformation

$$y_{if} = log(x_{if})$$

  - treat them as continuous ordinal data treat their rank as interval-scaled

Variables of Mixed Types

- A database may contain all the six types of variables
  - symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio
- One may use a weighted formula to combine their effects

$$d(i,j) = \frac{\Sigma_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\Sigma_{f=1}^{p} \delta_{ij}^{(f)}}$$

  - δij(f) = 0 if xif = xjf and f is asymmetric binary , or δij(f) = 1 otherwise

Vector Objects: Cosine Similarity

- Vector objects: keywords in documents, gene features in micro-arrays, …
- Applications: information retrieval, biologic taxonomy, ...
- Cosine measure: If d_1 and d_2 are two vectors, then
  $\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| \, ||d_2||$,
  where • indicates vector dot product, $||d||$: the length of vector d *(the Euclidean normal)*
  Cosine measure is a similarity measure: distance(d1,d2)=1-cos(d_1, d_2)
- Example:
  d_1 = 3 2 0 5 0 0 0 2 0 0
  d_2 = 1 0 0 0 0 0 0 1 0 2
  $d_1 \bullet d_2$ = 3*1+2*0+0*0+5*0+0*0+0*0+0*0+2*1+0*0+0*2 = 5
  $||d_1||$= (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)$^{0.5}$=(42)$^{0.5}$ = 6.481
  $||d_2||$ = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)$^{0.5}$=(6)$^{0.5}$ = 2.245
  cos(d_1, d_2) = .3150

Chapter 2. Cluster Analysis

1. What is Cluster Analysis?
2. Similarity/ dissimilarity
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods

Major Clustering Approaches (I)

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, ROCK, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

Chapter 2. Cluster Analysis

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes eventually the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* (MacQueen'67) and *k-medoids* (Kaufman & Rousseeuw'87) algorithms

The *K-Means* Clustering Method

Algorithm: ***k*-means.** The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

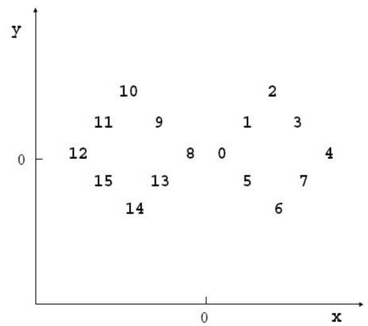
Method:

(1) arbitrarily choose *k* objects from *D* as the initial cluster centers;
(2) **repeat**
(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;
(5) **until** no change;

The *K-Means* Clustering example 1 (1/9)

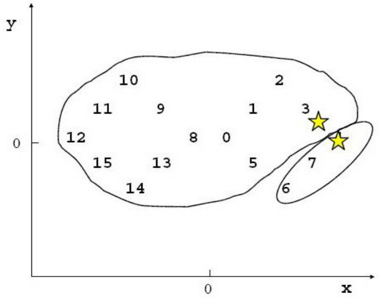
- Example ("Maschinelles Lernen and Data Mining" (page 3-11))

Id	x	y
0:	1.0	0.0
1:	3.0	2.0
2:	5.0	4.0
3:	7.0	2.0
4:	9.0	0.0
5:	3.0	-2.0
6:	5.0	-4.0
7:	7.0	-2.0
8:	-1.0	0.0
9:	-3.0	2.0
10:	-5.0	4.0
11:	-7.0	2.0
12:	-9.0	0.0
13:	-3.0	-2.0
14:	-5.0	-4.0
15:	-7.0	-2.0



The *K-Means* Clustering example 1 (2/9)

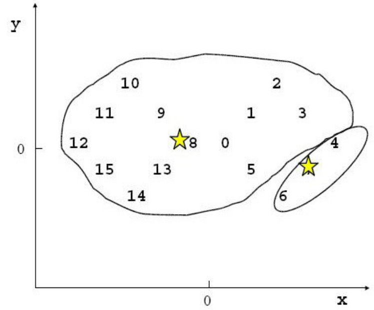
Clustering: (4 6 7) (0 1 2 3 5 8 9 10 11 12 13 14 15)
Cluster Centers: (7.0 -2.0) (-1.61538 0.46153)
Average Distance: 4.35887



The *K-Means* Clustering example 1 (3/9)

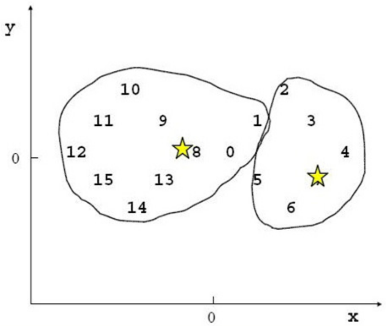
Clustering: (4 6 7) (0 1 2 3 5 8 9 10 11 12 13 14 15)
Cluster Centers: (7.0 -2.0) (-1.61538 0.46153)
Average Distance: 4.35887

Clustering: (2 3 4 5 6 7) (0 1 8 9 10 11 12 13 14 15)



The *K-Means* Clustering example 1 (4/9)

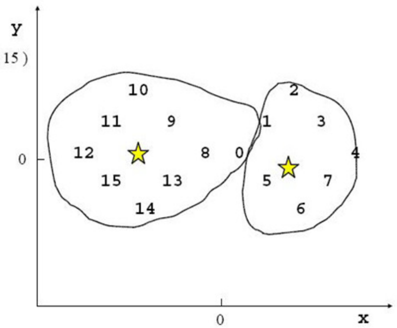
Clustering: (2 3 4 5 6 7) (0 1 8 9 10 11 12 13 14 15)
Cluster Centers: (6.0 -0.33334) (-3.6 0.2)
Average Distance: 3.6928



The *K-Means* Clustering example 1 (5/9)

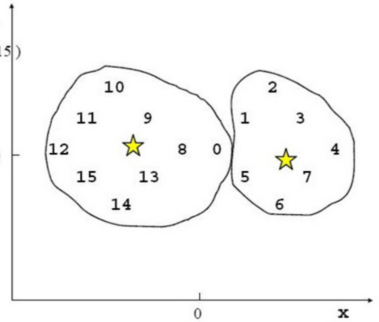
Clustering: (2 3 4 5 6 7) (0 1 8 9 10 11 12 13 14 15)
Cluster Centers: (6.0 -0.33334) (-3.6 0.2)
Average Distance: 3.6928

Clustering: (1 2 3 4 5 6 7) (0 8 9 10 11 12 13 14 15)



The *K-Means* Clustering example 1 (6/9)

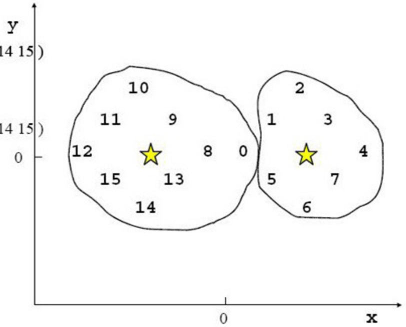
Clustering: (1 2 3 4 5 6 7) (0 8 9 10 11 12 13 14 15)
Cluster Centers: (5.57143 0.0) (-4.33334 0.0)
Average Distance: 3.49115



The *K-Means* Clustering example 1 (7/9)

Clustering: (1 2 3 4 5 6 7) (0 8 9 10 11 12 13 14 15)
Cluster Centers: (5.57143 0.0) (-4.33334 0.0)
Average Distance: 3.49115

Clustering: (0 1 2 3 4 5 6 7) (8 9 10 11 12 13 14 15)



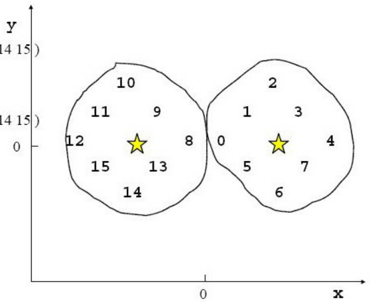
The *K-Means* Clustering example 1 (8/9)

Clustering: (4 6 7) (0 1 2 3 5 8 9 10 11 12 13 14 15)
Cluster Centers: (7.0 -2.0) (-1.61538 0.46153)
Average Distance: 4.35887

Clustering: (2 3 4 5 6 7) (0 1 8 9 10 11 12 13 14 15)
Cluster Centers: (6.0 -0.33334) (-3.6 0.2)
Average Distance: 3.6928

Clustering: (1 2 3 4 5 6 7) (0 8 9 10 11 12 13 14 15)
Cluster Centers: (5.57143 0.0) (-4.33334 0.0)
Average Distance: 3.49115

Clustering: (0 1 2 3 4 5 6 7) (8 9 10 11 12 13 14 15)
Cluster Centers: (5.0 0.0) (-5.0 0.0)
Average Distance: 3.41421



The *K-Means* Clustering example 1 (9/9)

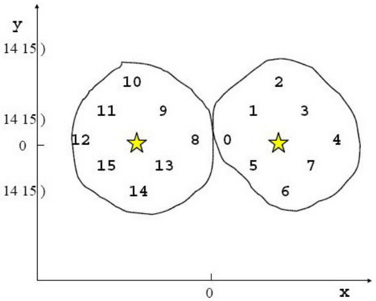
Clustering: (4 6 7) (0 1 2 3 5 8 9 10 11 12 13 14 15)
Cluster Centers: (7.0 -2.0) (-1.61538 0.46153)
Average Distance: 4.35887

Clustering: (2 3 4 5 6 7) (0 1 8 9 10 11 12 13 14 15)
Cluster Centers: (6.0 -0.33334) (-3.6 0.2)
Average Distance: 3.6928

Clustering: (1 2 3 4 5 6 7) (0 8 9 10 11 12 13 14 15)
Cluster Centers: (5.57143 0.0) (-4.33334 0.0)
Average Distance: 3.49115

Clustering: (0 1 2 3 4 5 6 7) (8 9 10 11 12 13 14 15)
Cluster Centers: (5.0 0.0) (-5.0 0.0)
Average Distance: 3.41421

Clustering: (0 1 2 3 4 5 6 7) (8 9 10 11 12 13 14 15)
No improvement.

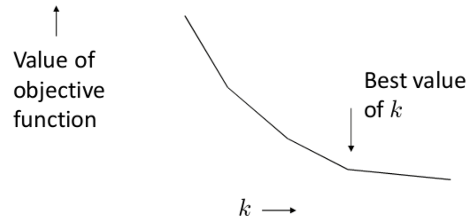


Comments on the *K-Means* Method

- Strength: *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, k, $t << n$.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Weakness
  - Applicable only when *mean* is defined, then what about categorical data?
  - Results can vary depending in initial random choices
  - Need to specify *k,* the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

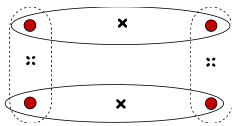
K-Means : A method for picking K

- Try different k, looking at the change in the value of the objective function as k increases
- This value falls rapidly until the right choice of k, and then changes less



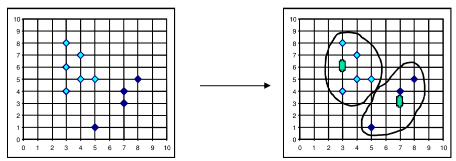
Variations of the *K-Means* Method

- Handling categorical data: *k-modes* (Huang'98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method



What Is the Problem of the K-Means Method?

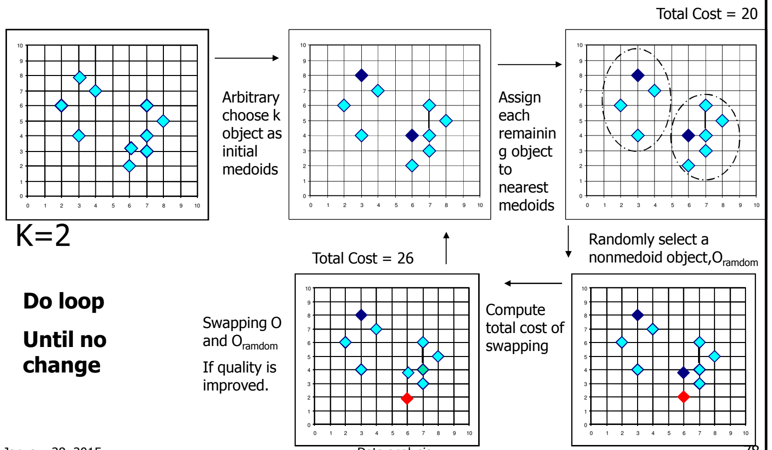
- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster. $E = \sum_{j=1}^{k} \sum_{p \in C_j} |p - o_j|,$



The *K-Medoids* Clustering Method

- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - *PAM* works effectively for small data sets, but does not scale well for large data sets
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling
- Focusing + spatial data structure (Ester et al., 1995)

A Typical K-Medoids Algorithm (PAM)



PAM (Partitioning Around Medoids) (Kaufman and Rousseeuw, 1987)

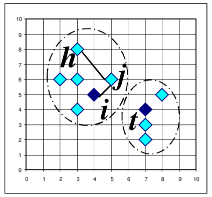
- Use real object to represent the cluster
  - Select ***k*** representative objects arbitrarily
    - Assign each object to the closest selected object
  - For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost ***TC***$_{ih}$
  - Pick the pair ***i***, ***h*** that offers the least swapping cost ***TC***$_{ih}$,
  - If $TC_{ih} < 0$, than
    - ***i*** is replaced by ***h***
    - assign each non-selected object to the most similar representative object
  - repeat steps 2-3 until no pair offers benefit

PAM Algorithm

Input:
  $D = \{t_1, t_2, ..., t_n\}$ // Set of elements
  A // Adjacency matrix showing distance between elements.
  k // Number of desired clusters.
Output:
  K // Set of clusters.
PAM Algorithm:
  arbitrarily select k medoids from D;
  repeat
    for each t_h not a medoid do
      for each medoid t_i do
        calculate TC_{ih};
    find i, h where TC_{ih} is the smallest;
    if $TC_{ih} < 0$ then
      replace medoid t_i with t_h;
  until $TC_{ih} \geq 0$;
  for each $t_i \in D$ do
    assign t_i to K_j where $dis(t_i, t_j)$ is the smallest over all medoids;

PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$

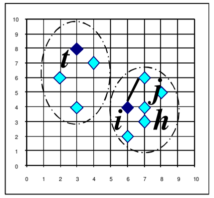
- **Case 1:** ***j*** currently belongs to ***i***. If ***i*** is replaced by ***h*** as a representative object and ***j*** is the closest to one of the other representative object ***t***, then ***j*** is reassigned to ***t***



$C_{jih} = d(j, t) - d(j, i)$

PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$

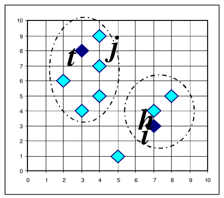
- **Case 2:** ***j*** currently belongs to the representative object ***i***. If ***i*** is replaced by ***h*** as a representative object and ***j*** is the closest to ***h*** then ***j*** is reassigned to ***h***



$C_{jih} = d(j, h) - d(j, i)$

PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$

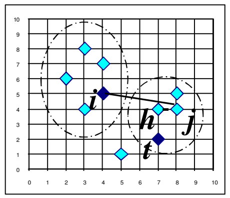
- **Case 3:** ***j*** currently belongs to the representative object ***t***. If ***i*** is replaced by ***h*** as a representative object and ***j*** is still closest to ***t*** then the assignment does not change.



$C_{jih} = 0$

PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$

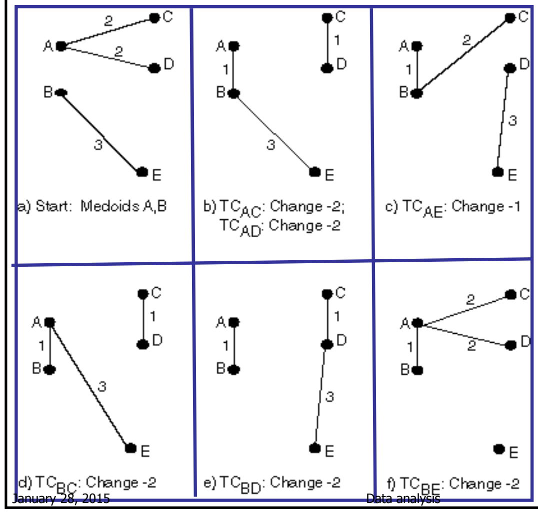
- **Case 4:** ***j*** currently belongs to the representative object ***t***. If ***i*** is replaced by ***h*** as a representative object and ***j*** is closest to ***h*** then ***j*** is reassigned to ***h.***



$C_{jih} = d(j, h) - d(j, t)$

PAM

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



What Is the Problem with PAM?

- Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- Pam works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(k(n-k)^2)$ for each iteration

    where n is # of data,k is # of clusters

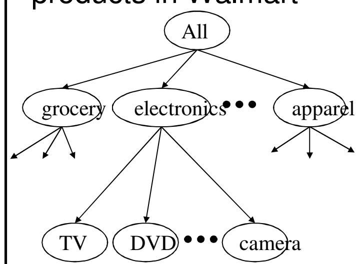
Chapter 2. Cluster Analysis

1. What is Cluster Analysis?
2. Similarity/ dissimilarity
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density based Methods

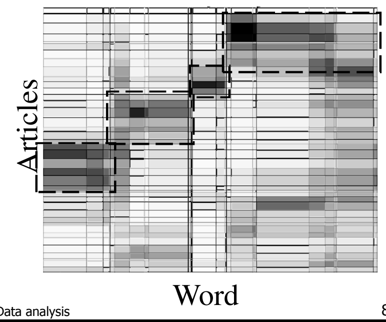
Observation 1: Hierarchical Structures

- Hierarchical structures often exist naturally among objects (e.g., taxonomy of animals)

A hierarchical structure of products in Walmart

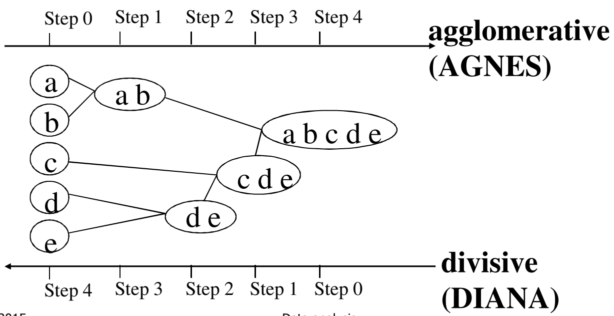


Relationships between articles and words (Chakrabarti, Papadimitriou, Modha, Faloutsos, 2004)



Hierarchical Clustering

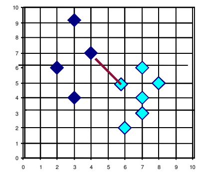
- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



Calculation of Distance between Clusters

- Single link: smallest distance between an element in one cluster and an element in the other

Minimum distance : $d_{min}(C_i, C_j) = \quad min_{p \in C_i,\, p' \in C_j} |p - p'|$

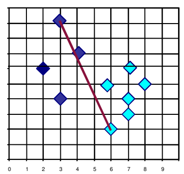


Single Linkage

Calculation of Distance between Clusters

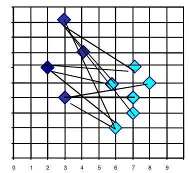
- Complete link: largest distance between an element in one cluster and an element in the other

Maximum distance : $d_{max}(C_i, C_j) = \quad max_{p \in C_i,\, p' \in C_j} |p - p'|$



Calculation of Distance between Clusters

- Average: avg distance between an element in one cluster and an element in the other



Average distance : $d_{avg}(C_i, C_j) = \quad \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p' \in C_j} |p - p'|$

Calculation of Distance between Clusters

- Centroid: distance between the centroids of two clusters

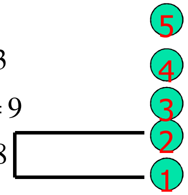
$$\text{Mean distance}: \quad d_{mean}(C_i, C_j) = \quad |\boldsymbol{m_i} - \boldsymbol{m_j}|$$

- Medoid: distance between the medoids of two clusters, i.e., dist(K_i, K_j) = dist(M_i, M_j)
  - Medoid: one chosen, centrally located object in the cluster

Example: single link

$$\begin{array}{c} \\ 1\\2\\3\\4\\5 \end{array}\begin{array}{c} \begin{array}{ccccc}1&2&3&4&5\end{array} \\ \left[\begin{array}{ccccc} 0&&&& \\ 2&0&&& \\ 6&3&0&& \\ 10&9&7&0& \\ 9&8&5&4&0 \end{array}\right]\end{array} \Rightarrow \begin{array}{c} \\ (1,2)\\3\\4\\5 \end{array}\begin{array}{c} \begin{array}{cccc}(1,2)&3&4&5\end{array} \\ \left[\begin{array}{cccc} 0&&& \\ 3&0&& \\ 9&7&0& \\ 8&5&4&0 \end{array}\right]\end{array}$$

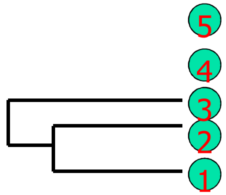
$$d_{(1,2),3} = \min\{d_{1,3}, d_{2,3}\} = \min\{6,3\} = 3$$
$$d_{(1,2),4} = \min\{d_{1,4}, d_{2,4}\} = \min\{10,9\} = 9$$
$$d_{(1,2),5} = \min\{d_{1,5}, d_{2,5}\} = \min\{9,8\} = 8$$



Example: single link

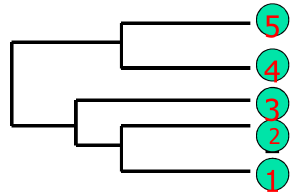
$$\begin{array}{c} \\ 1\\2\\3\\4\\5 \end{array}\begin{array}{c} \begin{array}{ccccc}1&2&3&4&5\end{array} \\ \left[\begin{array}{ccccc} 0&&&& \\ 2&0&&& \\ 6&3&0&& \\ 10&9&7&0& \\ 9&8&5&4&0 \end{array}\right]\end{array} \Rightarrow \begin{array}{c} \\ (1,2)\\3\\4\\5 \end{array}\begin{array}{c} \begin{array}{cccc}(1,2)&3&4&5\end{array} \\ \left[\begin{array}{cccc} 0&&& \\ 3&0&& \\ 9&7&0& \\ 8&5&4&0 \end{array}\right]\end{array} \Rightarrow \begin{array}{c} \\ (1,2,3)\\4\\5 \end{array}\begin{array}{c} \begin{array}{ccc}(1,2,3)&4&5\end{array} \\ \left[\begin{array}{ccc} 0&& \\ 7&0& \\ 5&4&0 \end{array}\right]\end{array}$$

$$d_{(1,2,3),4} = \min\{d_{(1,2),4}, d_{3,4}\} = \min\{9,7\} = 7$$
$$d_{(1,2,3),5} = \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8,5\} = 5$$



Example: single link

$$\begin{array}{c} \\ 1\\2\\3\\4\\5 \end{array}\begin{array}{c} \begin{array}{ccccc}1&2&3&4&5\end{array} \\ \left[\begin{array}{ccccc} 0&&&& \\ 2&0&&& \\ 6&3&0&& \\ 10&9&7&0& \\ 9&8&5&4&0 \end{array}\right]\end{array} \Rightarrow \begin{array}{c} \\ (1,2)\\3\\4\\5 \end{array}\begin{array}{c} \begin{array}{cccc}(1,2)&3&4&5\end{array} \\ \left[\begin{array}{cccc} 0&&& \\ 3&0&& \\ 9&7&0& \\ 8&5&4&0 \end{array}\right]\end{array} \Rightarrow \begin{array}{c} \\ (1,2,3)\\4\\5 \end{array}\begin{array}{c} \begin{array}{ccc}(1,2,3)&4&5\end{array} \\ \left[\begin{array}{ccc} 0&& \\ 7&0& \\ 5&4&0 \end{array}\right]\end{array}$$

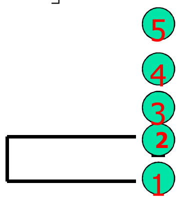


$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = 5$$

Example: complete link

$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & & & & \\
2 & 2 & 0 & & & \\
3 & 6 & 3 & 0 & & \\
4 & 10 & 9 & 7 & 0 & \\
5 & 9 & 8 & 5 & 4 & 0
\end{array}
\Rightarrow
\begin{array}{c|cccc}
 & (1,2) & 3 & 4 & 5 \\
\hline
(1,2) & 0 & & & \\
3 & 6 & 0 & & \\
4 & 10 & 7 & 0 & \\
5 & 9 & 5 & 4 & 0
\end{array}
$$

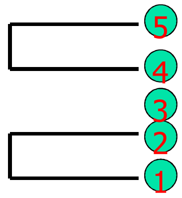
$$d_{(1,2),3} = \max\{d_{1,3}, d_{2,3}\} = \max\{6,3\} = 6$$
$$d_{(1,2),4} = \max\{d_{1,4}, d_{2,4}\} = \max\{10,9\} = 10$$
$$d_{(1,2),5} = \max\{d_{1,5}, d_{2,5}\} = \max\{9,8\} = 9$$



Example: complete link

$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & & & & \\
2 & 2 & 0 & & & \\
3 & 6 & 3 & 0 & & \\
4 & 10 & 9 & 7 & 0 & \\
5 & 9 & 8 & 5 & 4 & 0
\end{array}
\Rightarrow
\begin{array}{c|cccc}
 & (1,2) & 3 & 4 & 5 \\
\hline
(1,2) & 0 & & & \\
3 & 6 & 0 & & \\
4 & 10 & 7 & 0 & \\
5 & 9 & 5 & 4 & 0
\end{array}
\Rightarrow
\begin{array}{c|ccc}
 & (1,2) & 3 & (4,5) \\
\hline
(1,2) & 0 & & \\
3 & 6 & 0 & \\
(4,5) & 10 & 7 & 0
\end{array}
$$

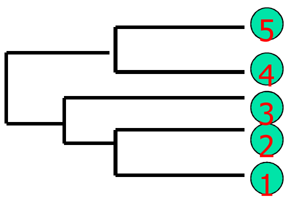
$$d_{(1,2),(4,5)} = \max\{d_{(1,2),4}, d_{(1,2),5}\} = \max\{10,9\} = 10$$
$$d_{3,(4,5)} = \max\{d_{3,4}, d_{3,5}\} = \max\{7,5\} = 7$$



Example: complete link

$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & & & & \\
2 & 2 & 0 & & & \\
3 & 6 & 3 & 0 & & \\
4 & 10 & 9 & 7 & 0 & \\
5 & 9 & 8 & 5 & 4 & 0
\end{array}
\Rightarrow
\begin{array}{c|cccc}
 & (1,2) & 3 & 4 & 5 \\
\hline
(1,2) & 0 & & & \\
3 & 6 & 0 & & \\
4 & 10 & 7 & 0 & \\
5 & 9 & 5 & 4 & 0
\end{array}
\Rightarrow
\begin{array}{c|ccc}
 & (1,2) & 3 & (4,5) \\
\hline
(1,2) & 0 & & \\
3 & 6 & 0 & \\
(4,5) & 10 & 7 & 0
\end{array}
$$

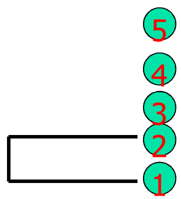
$$d_{(1,2,3),(4,5)} = \max\{d_{(1,2),(4,5)}, d_{3,(4,5)}\} = 10$$



Example: average link

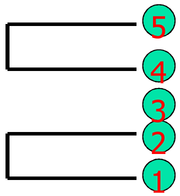
$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & & & & \\
2 & 2 & 0 & & & \\
3 & 6 & 3 & 0 & & \\
4 & 10 & 9 & 7 & 0 & \\
5 & 9 & 8 & 5 & 4 & 0
\end{array}
\Rightarrow
\begin{array}{c|cccc}
 & (1,2) & 3 & 4 & 5 \\
\hline
(1,2) & 0 & & & \\
3 & 4.5 & 0 & & \\
4 & 9.5 & 7 & 0 & \\
5 & 8.5 & 5 & 4 & 0
\end{array}
$$

$$d_{(1,2),3} = \frac{1}{2}(d_{1,3} + d_{2,3}) = \frac{6+3}{2} = 4.5$$
$$d_{(1,2),4} = \frac{1}{2}(d_{1,4} + d_{2,4}) = \frac{10+9}{2} = 9.5$$
$$d_{(1,2),5} = \frac{1}{2}(d_{1,5} + d_{2,5}) = \frac{9+8}{2} = 8.5$$



Example: average link

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & \\ 4.5 & 0 & & \\ 9.5 & 7 & 0 & \\ 8.5 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ (4,5) \end{array}\begin{array}{c} \begin{array}{ccc} (1,2) & 3 & (4,5) \end{array} \\ \begin{bmatrix} 0 & & \\ 4.5 & 0 & \\ 9 & 6 & 0 \end{bmatrix} \end{array}$$

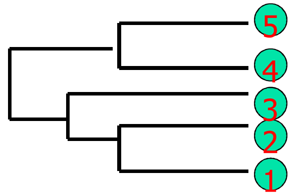


$$d_{(1,2),(4,5)} = \frac{1}{4}(d_{1,4} + d_{1,5} + d_{2,4} + d_{2,5}) = 9$$

$$d_{3,(4,5)} = \frac{1}{2}(d_{3,4} + d_{3,5}) = 6$$

Example: average link

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & \\ 4.5 & 0 & & \\ 9.5 & 7 & 0 & \\ 8.5 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ (4,5) \end{array}\begin{array}{c} \begin{array}{ccc} (1,2) & 3 & (4,5) \end{array} \\ \begin{bmatrix} 0 & & \\ 4.5 & 0 & \\ 9 & 6 & 0 \end{bmatrix} \end{array}$$



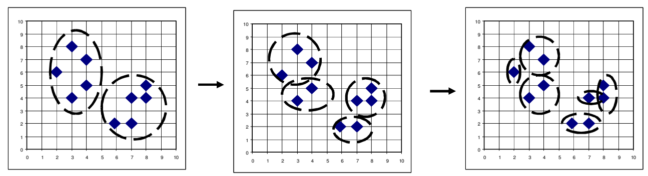
$$d_{(1,2,3),(4,5)} = \frac{1}{6}(d_{1,4} + d_{1,5} + d_{2,4} + d_{2,5} + d_{3,4} + d_{3,5}) = 8$$

Comparison of the Three Methods

- Single-link
  - "Loose" clusters
  - Individual decision, sensitive to outliers
- Complete-link
  - "Tight" clusters
  - Individual decision, sensitive to outliers
- Average-link
  - "In between"
  - Group decision, insensitive to outliers
- Which one is the best? Depends on what you need!

DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



DIANA --- Divisive Analysis

1. Initially, there is one large cluster consisting of all *n* objects.
2. Find the object, which has the highest average dissimilarity to all other objects. This object initiates a new cluster, S
3. For each object *i* inside the S, compute

   D_i= [average $d(i,j)$]$_{j \text{ not in S}}$ - [average $d(i,j)$]$_{j \text{ in S}}$

4. Find an object h for which the difference D_h is the lowest. If D_h is negative then h is, on the average close to the splinter group.
5. If D_h <w 0, then merge the object *h* to *S*

DIANA --- Divisive Analysis

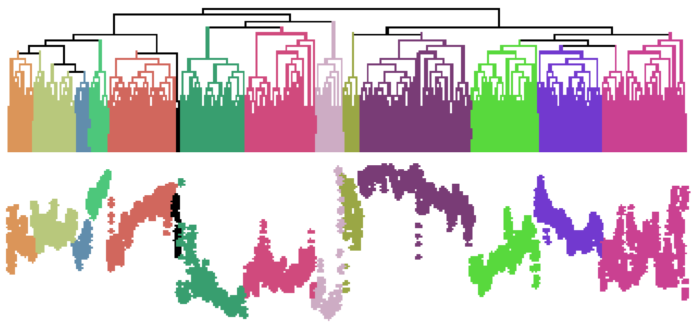
7. Repeat *Steps* 3 and 4 until all D_i are positif. The data set is then split into two clusters.
8. Select the cluster with the largest diameter. The diameter of a cluster is the largest dissimilarity between any two of its objects. Then divide this cluster, following steps 2-5.
9. Repeat *Step* 6 until all clusters contain only a single object.

Extensions to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
  - Do not scale well: time complexity of at least $O(n^2)$, where *n* is the number of total objects
  - Can never undo what was done previously
- Integration of hierarchical & distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - ROCK (1999): clustering categorical data by neighbor and link analysis
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

Problems with Dendogram

Messy to construct if number of points is large.



Chapter 2. Cluster Analysis

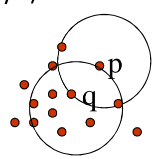
Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

Density-Based Clustering: Basic Concepts

- Two parameters:
  - *Eps*: Maximum radius of the neighbourhood
  - *MinPts*: Minimum number of points in an Eps-neighbourhood of that point
- $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid dist(p,q) \le Eps\}$
- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
  - *p* belongs to $N_{Eps}(q)$
  - core point condition:

    $|N_{Eps}(q)| \ge MinPts$

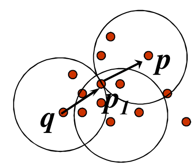


MinPts = 5

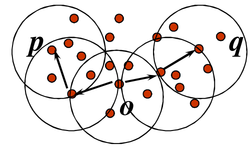
Eps = 1 cm

Density-Reachable and Density-Connected

- Density-reachable:
  - A point *p* is density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i

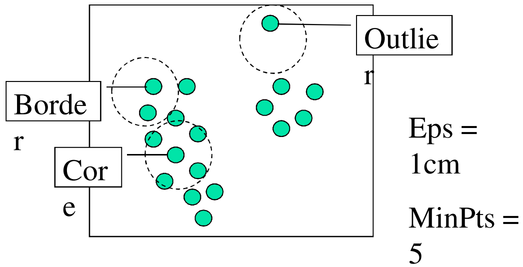


- Density-connected
  - A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*



DBSCAN: Density Based Spatial Clustering of Applications with Noise

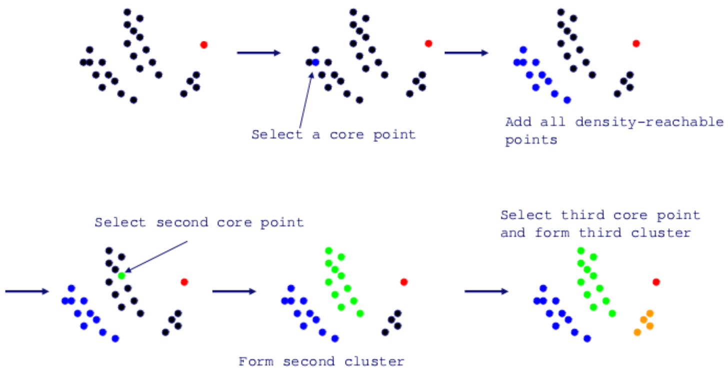
- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



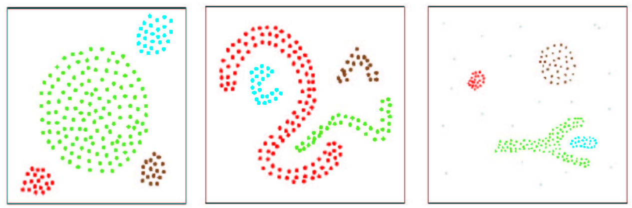
DBSCAN: The Algorithm

- Arbitrary select a point *p*
- If *p* is a core point, a cluster is formed: Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*.
- If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

DBSCAN Algorithm Revisited



DBSCAN



Arbitrary shape clusters found by DBSCAN

DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

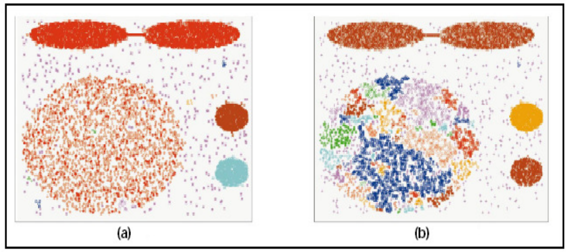
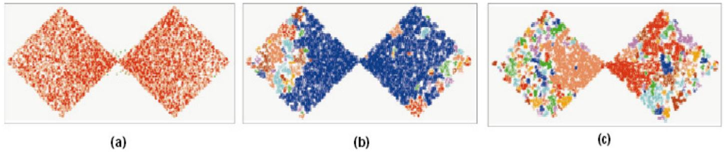
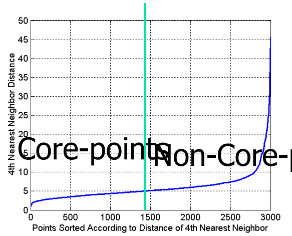


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



DBSCAN: Determining EPS and MinPts

- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor



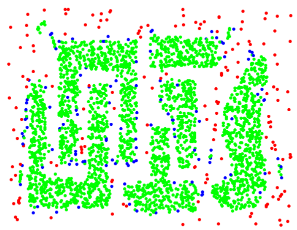
Run K-means for Minp=4 and not fixed

Complexity DBSCAN

- Time Complexity: $O(n^2)$—for each point it has to be determined if it is a core point, can be reduced to $O(n*\log(n))$ in lower dimensional spaces by using efficient data structures (n is the number of objects to be clustered);
- Space Complexity: O(n).

DBSCAN: Core, Border and Noise Points

Original Points

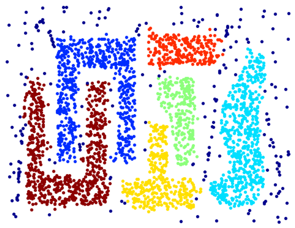


Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well

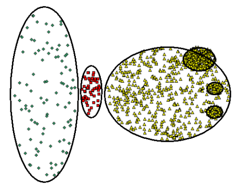
Original Points



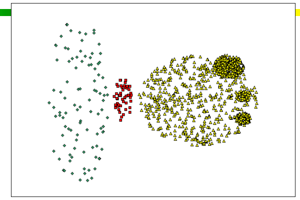
Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

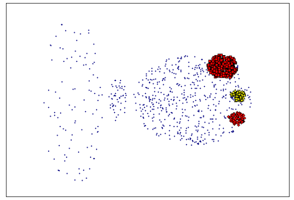
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

- Varying densities
- High-dimensional data

Data Analysis

Chapter 3 Principal Component Analysis

Mariem Gzara

Mastère professionnel en Informatique

Faculté des Sciences de Monastir

Chapter 3: Principal component analysis

- introduction
- Basics of statistics
- Basics of linear algebra
- Principal component analysis

Introducation

- PCA is a mathematical tool from applied linear algebra
- It is a simple, non-parametric method of extracting relevant information from confusing data sets
- It provides a raodmap for how to reduce a complex data set to a lower dimension

Basics of statistics

- Variance: a measure of the spread of the data in a data set with mean

$$\sigma^2 = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}{(n-1)}$$

- Varaiance- measure of the deviation from the mean for points in one dimension

Basics of statistics

- Covariance – a measure of how much each of the dimensions varies from the mean with respect to each other

$$\mathrm{var}(X) = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(X_i - \overline{X}\right)}{(n-1)}$$

- Covariance is measured between two dimensions

$$\mathrm{cov}(X,Y) = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{(n-1)}$$

to see if there is a relationship between the 2 dimensions, eg, number of hours studied and grade obtained

Basics of statistics

- What is the interpretation of covariance calculation?
- A positive value of covariance indicates that both dimensions increase or decrease together
- A negative value indicates while one increases the other decreases
- If covariance is zero: the tow dimensions are independent of each other

Basics of statistics

- Representing covariance among dimensions as a matrix, e.g., for 3 dimensions

$$C = \begin{bmatrix} \mathrm{cov}(X,X) & \mathrm{cov}(X,Y) & \mathrm{cov}(X,Z) \\ \mathrm{cov}(Y,X) & \mathrm{cov}(Y,Y) & \mathrm{cov}(Y,Z) \\ \mathrm{cov}(Z,X) & \mathrm{cov}(Z,Y) & \mathrm{cov}(Z,Z) \end{bmatrix}$$

- Properties:
  - Diagonal : variances of the variables
  - cov(X,Y)=cov(Y,X), hence matrix is symmetrical about the diagonal (upper triangular)
  - M-dimensional data will result in m×m covariance matrix

Basics of linear algebra

- Matrix A:

$$A = [a_{ij}]_{m\times n} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

- Matrix transpose

$$B = [b_{ij}]_{n\times m} = A^T \Leftrightarrow b_{ij} = a_{ji}; \quad 1 \le i \le n, 1 \le j \le m$$

- Vector a

$$a = \begin{bmatrix} a_1 \\ \dots \\ a_n \end{bmatrix}; \quad a^T = [a_1, \dots, a_n]$$

Basics of linear algebra

- Matrix Multiplication

$$A = [a_{ij}]_{m\times p}; B = [b_{ij}]_{p\times n};$$
$$AB = C = [c_{ij}]_{m\times n}, where\ c_{ij} = row_i(A) \cdot col_j(B)$$

- Outer vector product $a = A = [a_{ij}]_{m\times 1}; b^T = B = [b_{ij}]_{1\times n};$

$$c = a \times b = AB, an\ m \times n\ matrix$$

- Vector-matrix product

$$A = [a_{ij}]_{m\times n}; b = B = [b_{ij}]_{n\times 1};$$
$$C = Ab = an\ m \times 1\ matrix = vector\ of\ length\ m$$

Basics of linear algebra

- Inner (dot) product: $a^T \cdot b = \sum_{i=1}^{n} a_i b_i$
- Length (eucledian norm) of a vector

$$\|a\| = \sqrt{a^T \cdot a} = \sqrt{\sum_{i=1}^{n} a_i^2}$$

- a is normalized iff ||a||=1
- The angle between two n-dimensioanl vectors

$$\cos\theta = \frac{a^T \cdot b}{||a|| ||b||}$$

Basics of linear algebra

- An inner product is a measure of collinearity:
  - a and b are orthogonal iff $a^T.b=0$
  - a and b are collinear iff $a^T.b=||a||\ ||b||$
- A set of vectors is linearly independent if no vector is a linear combination of other vectors
- Trace

$$A=[a_{ij}]_{n\times n};\ tr[A]=\sum_{j=1}^{n} a_{jj}$$

Basics of linear algebra

- Determinant

$$A=[a_{ij}]_{n\times n};$$

$$\det(A)=\sum_{j=1}^{n} a_{ij}A_{ij};\quad i=1,\ldots,n;$$

$$A_{ij}=(-1)^{i+j}\det(M_{ij})$$

$$\det(AB)=\det(A)\det(B)$$

Basics of linear algebra

- A ($n\times n$) is nonsingular if there exists B such that :

  $A\,B=B\,A=I_n$; $B=A^{-1}$
- A=[2 3 ; 2 2] , B=[-1 3/2 ; 1 -1]
- A is nonsingular if $||A||\neq 0$
- Pseudo-inverse for a non square matrix, provided

  A^TA is not singular

  $A^{\#}=[A^TA]^{-1}A^T$

  $A^{\#}A=I$

Basics of linear algebra

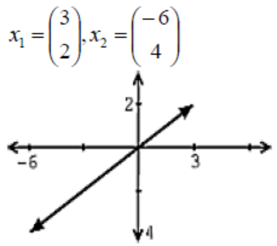
- A set of n-dimensional vectors $x_i \in R^n$, are said to be linearly independent if none of them can be written as a linear combination of the others
- In other words

  $c_1x_1+c_2x_2+\ldots+c_kx_k=0$

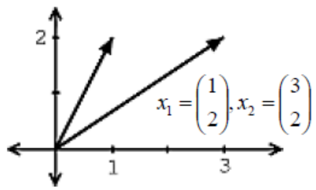
  Iff $c_1=c_2=\ldots=c_k=0$

Basics of linear algebra

- Another approach to reveal a vectors independence is by graphing the vectors.



Not linearly independent vectors



Linearly independent vectors

Basics of linear algebra

- Span: a span of a set of vectors $x_1, x_2, \ldots, x_k$ is the set of vectors that can be written as a linear combination of $x_1, x_2, \ldots, x_k$

  $\text{Span}(x_1, x_2, \ldots, x_k)=$

  $\{c_1x_1+c_2x_2+ \ldots +c_kx_k | c_1, c_2, \ldots, c_k \in \Re\}$

Basics of linear algebra

- A basis for R^n is a set of vectors which:
  - Span R^n, i.e. any vector in this n-dimensional space can be written as linear combination of these basis vectors
  - Are linearly independent
- Clearly, any set of n-linearly independent vectors form basis vectors for R^n

Basics of linear algebra

- An orthogonal basis of a vector space V with an inner product, is a set of basis vectors whose elements are mutually orthogonal and a magnitude 1 (unit vectors)
- Elements is an orthogonal basis do not have to be unit vectors, but must be mutually perpendicular. It is easy to change the vectors in an orthogonal basis, and indeed this is a typical way that an orthogonal basis is constructed

Basics of linear algebra

- Two vectors are orthogonal if they are perpendicular, i.e., they form a right angle, i.e., if their inner product is zero

$$a^T \cdot b = \sum_{i=1}^{n} a_i b_i = 0 \quad \Rightarrow \quad a \perp b$$

- The standard basis of the n-dimensioanl Euclidean space R^n is an example of orthonormal (and ordered basis)

Basics of linear algebra

- Eigenvalue problem: the eigenvalue problem is any problem having the following form

  A.v=λ.v

  A: m×m matrix

  v: m ×1 non-zero vector

  λ: scalar

- Any value of λ for which this equation has a solution is called the eigenvalue of A and the vector v which corresponds to this value is called the eigenvector of A

Basics of linear algebra

- Consider the following example:

$$\underbrace{\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}}_{A} \underbrace{\times}_{.} \underbrace{\begin{bmatrix} 3 \\ 2 \end{bmatrix}}_{v} = \begin{bmatrix} 12 \\ 8 \end{bmatrix} \underbrace{=}_{=} \underbrace{4}_{\lambda} \underbrace{\times}_{.} \underbrace{\begin{bmatrix} 3 \\ 2 \end{bmatrix}}_{v}$$

- Therefore, (3,2) is an eigenvector of the square matrix A and 4 is an eigenvalue of A

Basics of linear algebra

- Scale vector (3,2) by a value 2 to get (6,4)

$$2 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 6 \\ 4 \end{bmatrix} = \begin{bmatrix} 24 \\ 16 \end{bmatrix} = 4 \times \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

- A vector consists of both length and direction. Scaling a vector changes only its length and not its direction.

Basics of linear algebra
Calculating eigenvectors and eigenvalues

- Given matrix A, how can we calculate the eigenvector and eigenvalues for A?
- Calculating eigenvectors and eigenvalues
  - Simple matrix algebra shows that:

    $A.v=\lambda.v \quad \Leftrightarrow \quad A.v-\lambda.I.v=0 \quad \Leftrightarrow \quad (A-\lambda.I).v=0$
  - Finding the roots of $|A-\lambda.I|$ will give the eigenvalues and for each of these eigenvalues there will be an eigenvector

Basics of linear algebra
Calculating eigenvectors and eigenvalues

- Let $A=\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$
- Then

$$|A-\lambda.I| = \left|\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - \lambda\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right| = \left|\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right|$$

$$= \left|\begin{bmatrix} -\lambda & 1 \\ -2 & -3-\lambda \end{bmatrix}\right| = (-\lambda\times(-3-\lambda))-(-2\times 1) = \lambda^2+3\lambda+2$$

- And setting the determinant to 0, we obtain 2 eigenvalues: $\lambda_1=-1$ and $\lambda_2=-2$

Basics of linear algebra
Calculating eigenvectors and eigenvalues

- For λ_1 the eigenvector is:

$$(A-\lambda_1.I).v_1=0$$

$$\begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix}.\begin{bmatrix} v_{1:1} \\ v_{1:2} \end{bmatrix}=0$$

$$v_{1:1}+v_{1:2}=0 \quad and \quad -2v_{1:1}-2v_{1:2}=0$$

$$v_{1:1}=-v_{1:2}$$

- Therefore the first eigenvector is any column vector in which the two elements have equal magnitude and opposite sign

Basics of linear algebra
Calculating eigenvectors and eigenvalues

- Therefore eigenvector v_1 is

$$v_1=k_1\begin{bmatrix} +1 \\ -1 \end{bmatrix}$$

where k_2 is some constant

- Similarly we find that eigenvector v_2

$$v_2=k_2\begin{bmatrix} +1 \\ -2 \end{bmatrix}$$

where k_2 is some constant

Basics of linear algebra

Properties eigenvectors and eigenvalues

- Eigenvectors can only be found for square matrices and not every square matrix has eigenvectors.
- Given an $m \times m$ matrix (with eigenvectors), we can find n eigenvectors
- All eigenvectors of a symmetric matrix are perpendicular to each other, no matter how many dimensions we have
- In practice eigenvectors are normalized to have unit length

Principal Component Analysis Exemple of a problem

- We collected m parameters about n=100 students:
  - height,
  - weight,
  - hair color,
  - average garde, …
- We want to find the most important parameters that best describe a student

Principal Component Analysis Exemple of a problem

- Each student has a vector of data which describes him of length m
  - (example (180,70,'purple',84, …)
- We have n=100 such vectors. Let's put them in one matrix, where each column is one student vector
- We have a $m \times n$ matrix. This will be the input of our problem

Principal Component Analysis Exemple of a problem

- Which parameter can we ignore?
  - Constant parameter (number of heads)
    - 1,1, …,1.
  - Constant parameter with some noise – (thikness of hair)
    - 0.003, 0.005, 0.002, …, 0.008 → low variance
  - Parameter that is linearly dependent on other parameters (head size and height)
    - $Z=aX+bY$

Principal Component Analysis
Exemple of a problem

- Wich parameters do we want to keep?
  - Parameter that doesn't depend on others (e.g. eye color), i.e. uncorrelated →low covariance
  - Parameter that changes a lot (grades)
  - High variance

Principal Component Analysis

- Questions
  - How we describe most important features using math?
    Variance
  - How do we represent our data so that the most important features can be extracted easily?
    Change of basis

Change of basis

- Let X and Y be m×n matrices related by a linear transformation P
- X is the original recorded data set and Y is a re-representation of that data set

PX=Y

Let's define;

P[i,] the i^{th} row of P

x[,i] the i^{th} column of X

y[,i] the i^{th} column of Y

We have

y[,i]= P[i,] × x[,i]

Change of basis!!!

- X is the original recorded data set
- The rows of P, {p1, p2, …, pm} are a set of new basis vectors for expressing the columns of X
- Y is the representation of the data set X in the new basis vectors P= {p1, p2, …, pm}
  - P is a matrix that transforms X into Y
  - PX=Y

Geometrically, P is a rotation and a stretch (scaling) which again transforms X into Y

Change of basis !!!

- Lets write out the explicit dot products of PX

$$PX = \begin{bmatrix} p_1 \\ \vdots \\ p_m \end{bmatrix} \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}$$

- We can note the form of each column of Y

$$Y = \begin{bmatrix} p_1 \cdot x_1 & \cdots & p_1 \cdot x_n \\ \vdots & \ddots & \vdots \\ p_m \cdot x_1 & \cdots & p_m \cdot x_n \end{bmatrix}$$

- We can set that each coefficient of yi is a dot-product of xi with the corresponding row in P

$$y_i = \begin{bmatrix} p_1 \cdot x_i \\ \vdots \\ p_m \cdot x_i \end{bmatrix}$$

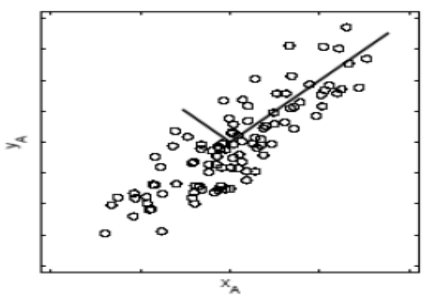
- In other words, the jth coefficient of yi is a projection onto the jth row of P
- Therfore, the rows of P are a new set of basis vectors for representing the columns of X

Change of basis!!!

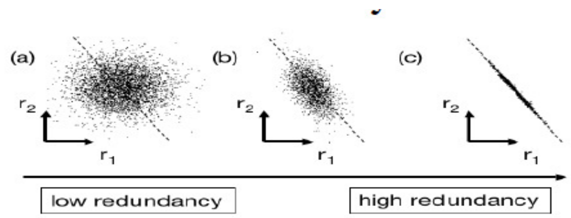
- Changing the basis doesn't change the data –only its representation
- Changing the basis is actually projecting the data vectors on the basis vectotrs
- Geometrically, P is a rotation and a stretch of X
- If P basis is orthonormal (length=1) then the transformation P is only a rotation

Change of basis !!!

- An exemple of change of basis



Change of basis



- Multiple sensors record the same dynamic information
- Consider a range of possible plots between two arbitrary measurment types r1 and r2
- Panel(a) depicts two recordings with no redundancy, i.e., they are un-correlated, e.g., person's height and his GPA
- However, in panel (c) both recordings appear to be strongly related, i.e. one can be expressed in terms of the other

PCA Process

1. Substruct the mean from each of the dimensions
   - This produces a data set whose mean is zero
   - Subtracting the mean makes varaiance and covariance calculation easier by simplifying their equations.
   - The variance and co-varaince values are not affected by the mean value.
2. Calculate the covariance matrix
3. Calculate the eigenvectors and eigenvalues of the covariance matrix
   - Since the covariance matrix is symmetwe, the eigenvectors are orthogonal.
4. Order the eigenvalues, highest tol owest. This gives the components in order of significance.

   $(\lambda_1 , \lambda_2, \ldots, \lambda_m)$

PCA Process

5. Derive the new data

FinalData=RowFeatureVectorxRowZeroMeanData

- RowFeatureVector is the matrix with the eigenvectors in the columns transposed so that the eigenvectors are now in the rows, with the most significant eigenvector at the top.

RowZeroMeanData

- The data items are in each column ,with each row holding a separate dimension

Dimensionality reduction

When the λi's are sorted in descending order, the proportion of variance explained by the *r -first principal components is:*

$$\frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{m} \lambda_i} = \frac{\lambda_1 + \lambda_2 + \ldots + \lambda_r}{\lambda_1 + \lambda_2 + \ldots + \lambda_p + \ldots + \lambda_m}$$

If we reduce the dimensionality (i.e.,*r<m), we choose the r -first principal components that give a proportion of variance higher than a given threshold*

PCA Process

X_1	X_2		X'_1	X'_2
2.5	2.4		0.69	0.49
0.5	0.7		-1.31	-1.21
2.2	2.9		0.39	0.99
1.9	2.2		0.09	0.29
3.1	3.0	$\Rightarrow$ $\overline{X_1} = 1.81$, $\overline{X_2} = 1.91$ $\Rightarrow$	1.29	1.09
2.3	2.7		0.49	0.79
2.0	1.6		0.19	-0.31
1.0	1.1		-0.81	-0.81
1.5	1.6		-0.31	-0.31
1.2	0.9		-0.71	-1.01

Mean of attribute 1

Mean of attribute 2

Original recorded data set: 2 attributes, 10 data

Substruct the mean from each of the dimensions

PCA Process

- Covariance matrix

$$\mathrm{cov} = \begin{bmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{bmatrix}$$

- Eignevalues and eigenvectors

$$eigenvalues = \begin{bmatrix} 0.490833989 \\ 1.28402771 \end{bmatrix}$$

$$eigenvectors = \begin{bmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{bmatrix}$$

PCA Process

- Order the eigenvalues, highest tol owest. This gives the components in order of significance. eigenvazlues [1.28402771 0.490833989]

$$\begin{bmatrix} -0.677873399 & -0.735178656 \\ -0.735178656 & 0.677873399 \end{bmatrix}$$

PCA Process

$$\begin{bmatrix} -0.677873399 & -0.735178656 \\ -0.735178656 & 0.677873399 \end{bmatrix} \times \begin{array}{cc} X_1 & X_2 \\ 2.5 & 2.4 \\ 0.5 & 0.7 \\ 2.2 & 2.9 \\ 1.9 & 2.2 \\ 3.1 & 3.0 \\ 2.3 & 2.7 \\ 2.0 & 1.6 \\ 1.0 & 1.1 \\ 1.5 & 1.6 \\ 1.2 & 0.9 \end{array}^{T} = \begin{array}{cc} newX_1 & newX_2 \\ -0.827870186 & -0.175115307 \\ 1.77758033 & 0.142857227 \\ -0.992197494 & 0.384374989 \\ -0.274210416 & 0.130417207 \\ -1.67580142 & -0.209498461 \\ -0.912949103 & 0.175282444 \\ 0.0991094375 & -0.349824698 \\ 1.14457216 & 0.0464172582 \\ 0.438046137 & 0.0177646297 \\ 1.22382956 & -0.162675287 \end{array}$$

Final Data in the new basis

References (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD'99.
- M. M. Breunig, H.-P. Kriegel, R. Ng, J. Sander. LOF: Identifying Density-Based Local Outliers. SIGMOD 2000.
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. VLDB'98.
- V. Ganti, J. Gehrke, R. Ramakrishan. CACTUS Clustering Categorical Data Using Summaries. KDD'99.

References (2)

- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB'98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. In *ICDE'99*, pp. 512-521, Sydney, Australia, March 1999.
- A. Hinneburg, D.l A. Keim: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. KDD'98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.
- G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *COMPUTER*, 32(8): 68-75, 1999.
- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98.

References (3)

- G. J. McLachlan and K.E. Bkasford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- P. Michaud. Clustering Techniques. Future Generation Computer Systems, 13, 1997.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- L. Parsons, E. Haque and H. Liu, Subspace Clustering for High Dimensional Data: A Review, SIGKDD Explorations, 6(1), June 2004
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition,.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB'98.
- A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-Based Clustering in Large Databases, ICDT'01.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB'97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : An efficient data clustering method for very large databases. SIGMOD'96.
- **A Tutorial on Principal Component Analysis Aly A. Farag Shireen Elhabian University of Louisville, CVIP Lab September 2009**

References

- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley, 2003
- T. Dasu, T. Johnson, S. Muthukrishnan, V. Shkapenyuk. Mining Database Structure; Or, How to Build a Data Quality Browser. SIGMOD'02
- H. V. Jagadish et al., Special Issue on Data Reduction Techniques. Bulletin of the Technical Committee on Data Engineering, 20(4), Dec. 1997
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999
- E. Rahm and H. H. Do. Data Cleaning: Problems and Current Approaches. *IEEE Bulletin of the Technical Committee on Data Engineering. Vol.23, No.4*
- Data Mining: Concepts and Techniques (3rd ed.) Jiawei Han, Micheline Kamber, and Jian Pei University of Illinois at Urbana-Champaign & Simon Fraser University, 2009 Han, Kamber & Pei.
- DISCOVERING KNOWLEDGE IN DATA: An Introduction to Data Mining, DANIEL T. LAROSE, A JOHNWILEY& SONS,INC.,PUBLICATION, 2005.
- Stéphane Tufféry, Data mining et statistique décisionnelle, éditions TECHNIP, 2010