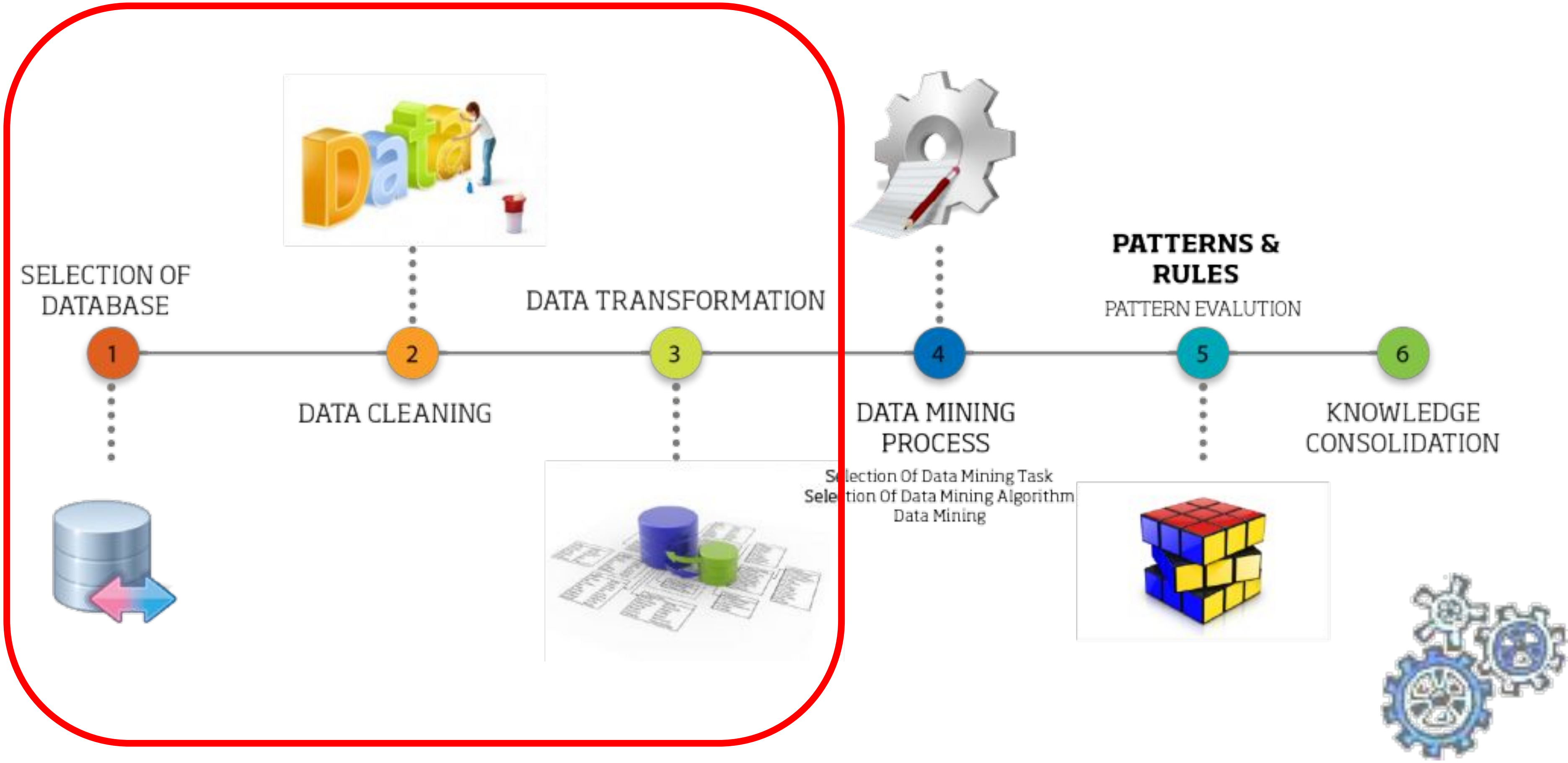
Data Preparation

KDD Process



SELECTION OF DATABASE

DATA CLEANING

DATA TRANSFORMATION

DATA MINING PROCESS

Selection Of Data Mining Task
Selection Of Data Mining Algorithm
Data Mining

PATTERNS & RULES

PATTERN EVALUTION

KNOWLEDGE CONSOLIDATION

Data understanding vs Data preparation

Data understanding provides general information about the data like

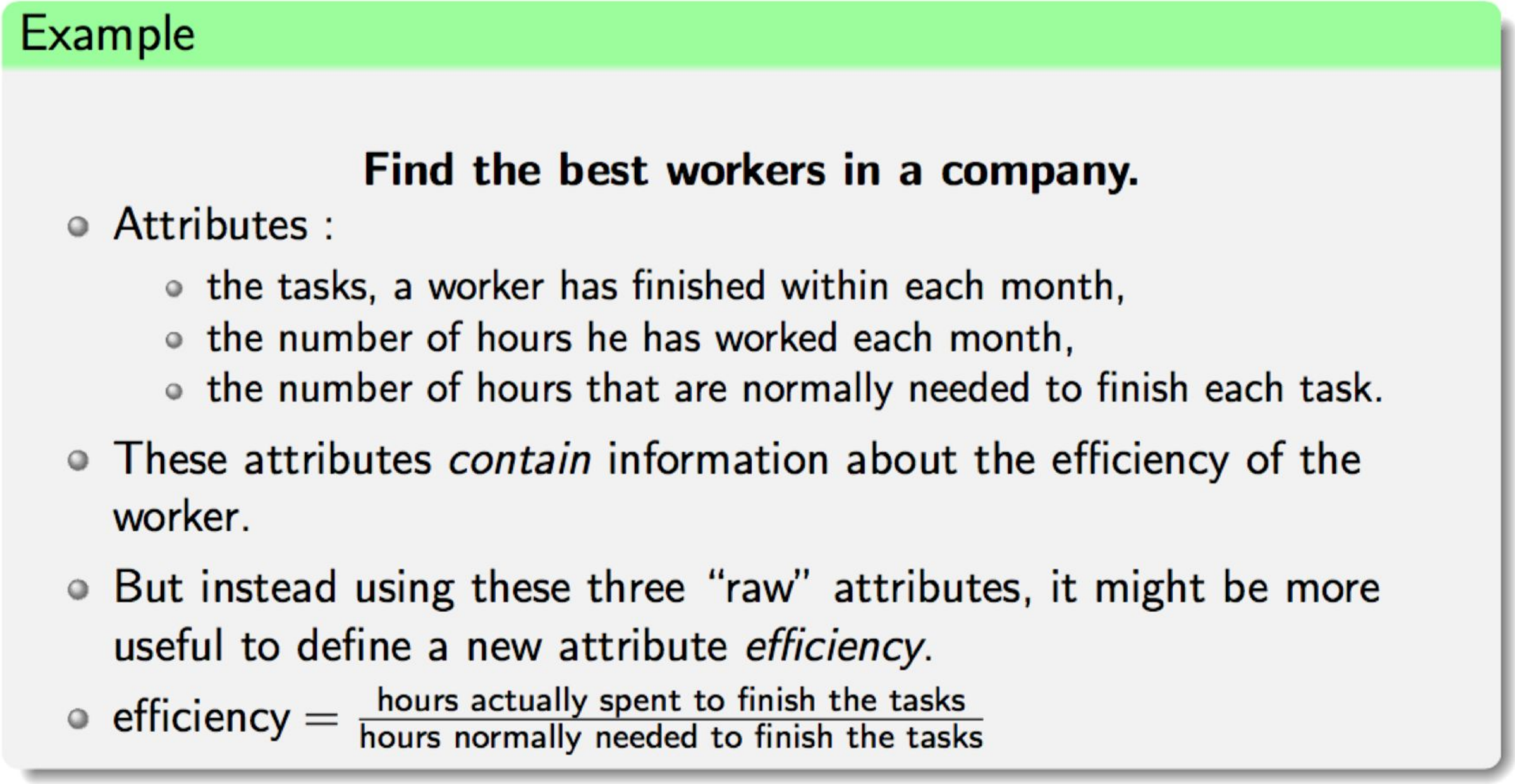
- The existence of **missing values**
- The existence of **outliers**
- the character of attributes
- **dependencies** between attributes.

Data preparation uses this information to

- select attributes,
- reduce the dimension of the data set,
- select records,
- treat missing values,
- treat outliers,
- integrate, unify and transform data
- improve data quality

Feature Extraction

Construct (new) features from the given attributes



Example

Find the best workers in a company.

- Attributes :
  - the tasks, a worker has finished within each month,
  - the number of hours he has worked each month,
  - the number of hours that are normally needed to finish each task.
- These attributes *contain* information about the efficiency of the worker.
- But instead using these three "raw" attributes, it might be more useful to define a new attribute *efficiency*.
- $\text{efficiency} = \dfrac{\text{hours actually spent to finish the tasks}}{\text{hours normally needed to finish the tasks}}$

Data Reduction

Reducing the amount of data

- Reduce the number of **records**
  - Data Sampling
  - Clustering
- Reduce the number of **columns** (attributes)
  - Select a subset of attributes
  - Generate a new (a smaller) set of attributes

Sampling

- Improve the execution time of data mining algorithms

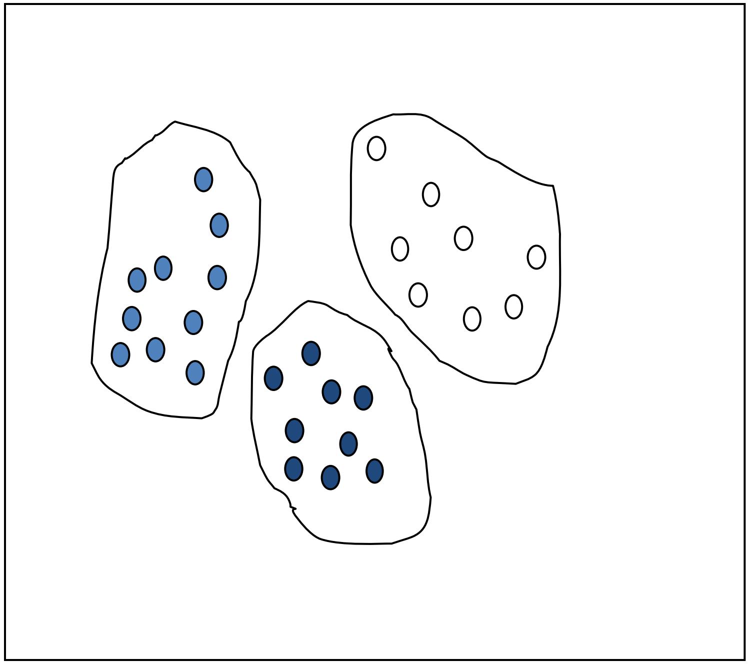
- **Problem**: how to select a subset of **representative** data?

  - **Random sampling:** it can generate problems due to the possible peaks in the data

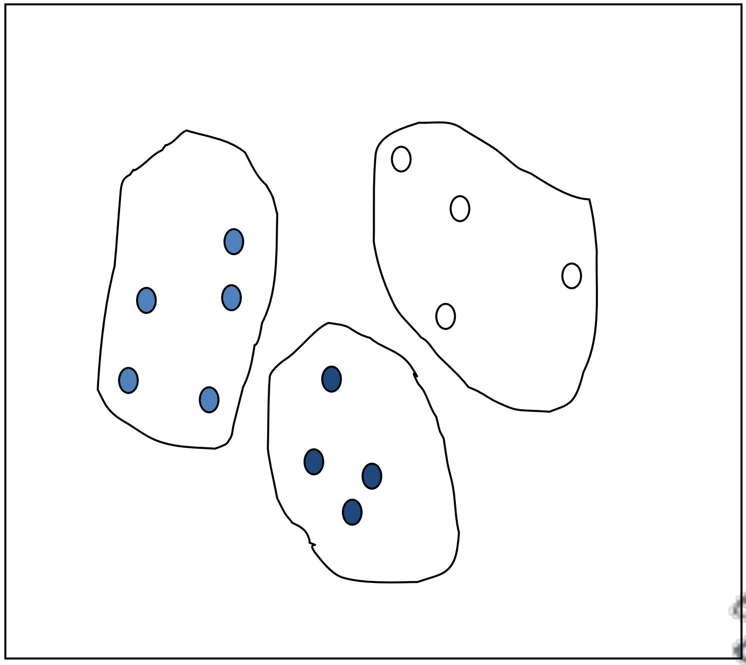
  - **Stratified sampling**:
    - Approximation of the percentage of each class
    - Suitable for distribution with peaks: each peak is a **layer**

Stratified Sampling

Raw Data



Cluster/Stratified Sample



Reduction of Dimensionality

- **Selection of a subset of attributes** that is as small as possible and sufficient for the data analysis.
  - removing (more or less) **irrelevant** features
  - removing **redundant** features.

Removing irrelevant/redundant features

- For **removing irrelevant features**, a performance measure is needed that indicates how well a feature or subset of features performs w.r.t. the considered data analysis task

- For removing **redundant features**, either a performance measure for subsets of features or a correlation measure is needed.

Reduction of Dimensionality

Manual

- After analyzing the **significance** and/or **correlation** with other attributes

Automatic: Selecting the top-ranked features

- Incremental Selection of the “best” attributes
- “Best” = with respect to a specific measure of statistical significance (e.g.: information gain).

Feature Selection Techniques

- **Selecting the top-ranked features**: Choose the features with the best evaluation when single features are evaluated.

- **Selecting the top-ranked subset:** Choose the subset of features with the best performance. This requires exhaustive search and is impossible for larger numbers of features. (For 20 features there are already more than one million possible subsets.)

- **Forward selection:** Start with the empty set of features and add features one by one. In each step, add the feature that yields the best improvement of the performance.

- **Backward elimination:** Start with the full set of features and remove features one by one. In each step, remove the feature that yields to the least decrease in performance.

Data Cleaning

- How to handle anomalous values
- How to handle outliers
- Data Transformations

Anomalous Values

- **Missing values**
  - NULL, ?
- **Unknown Values**
  - Values without a real meaning
- **Not Valid Values**
  - Values not significant

Manage Missing Values

1. Elimination of records
2. Substitution of values

Note: it can influence the original distribution of numerical values

- Use mean/median/mode
- Estimate missing values **using the probability distribution** of existing values
- Data Segmentation and using mean/mode/median of each **segment**
- Data Segmentation and using **the probability distribution within the segment**
- Build a model of **classification/regression** for computing missing values

Data Transformation: Motivations

- Data with errors and incomplete

- Data not adequately distributed
  - Strong asymmetry in the data
  - Many peaks

- Data transformation can reduce these issues

Goals

- Define a transformation T on the attribute X:

$$Y = \mathrm{T}(X)$$

such that :

  - Y preserve the **relevant** information of X
  - Y eliminates at least one of the problems of X
  - Y is more **useful** of X

Goals

- **Main goals:**
  - stabilize the variances
  - normalize the distributions
  - Make linear relationships among variables
- **Secondary goals:**
  - simplify the elaboration of data containing features you do not like
  - represent data in a scale considered more suitable

Why linear correlation, normal distributions, etc?

- Many statistical methods require linear correlations, normal distributions, the absence of outliers

- Many data mining algorithms have the ability to automatically treat **non-linearity** and **non-normality**
  - The algorithms work still better if such problems are treated

Normalizations

- min-max normalization

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- z-score normalization

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

- normalization by decimal scaling

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

Normalization by decimal scaling

- Suppose that the recorded values of F range from − 986 to 917.
- The maximum absolute value of F is 986.
- To normalize by decimal scaling, we therefore divide each value by 1,000 (i.e., j = 3) so that −986 normalizes to −0.986 and 917 normalizes to 0.917.

Methods

- Exponential transformation

$$T_p(x) = \begin{cases} ax^p + b & (p \neq 0) \\ c\log x + d & (p = 0) \end{cases}$$

- with a,b,c,d and p real values
  - Preserve the order
  - Preserve some basic statistics
  - They are continuous functions
  - They are derivable
  - They are specified by simple functions

Better Interpretation

- Linear Transformation

  1€ = 1936.27 Lit.

  - *p=1, a= 1936.27 ,b =0*

  ºC= 5/9(ºF -32)

  - *p = 1, a = 5/9, b = -160/9*

$$T_p(x) = \begin{cases} ax^p + b & (p \neq 0) \\ c \log x + d & (p = 0) \end{cases}$$

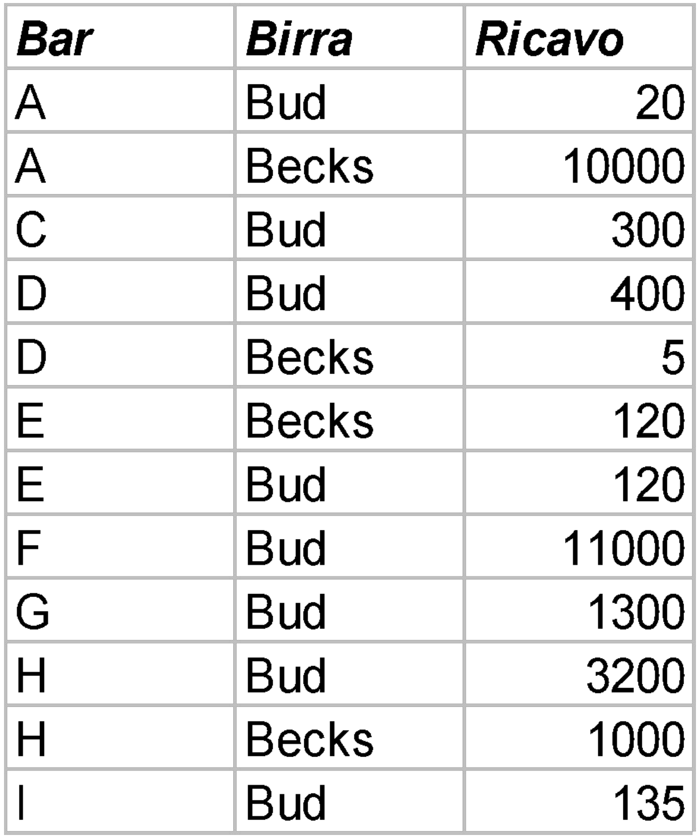
Stabilizing the Variance

- **Logarithmic Transformation**

$$T(x) = c\log x + d$$

  - Applicable to positive values
  - Makes homogenous the variance in log-normal distributions
    - E.g.: normalize seasonal peaks

Logarithmic Transformation: Example

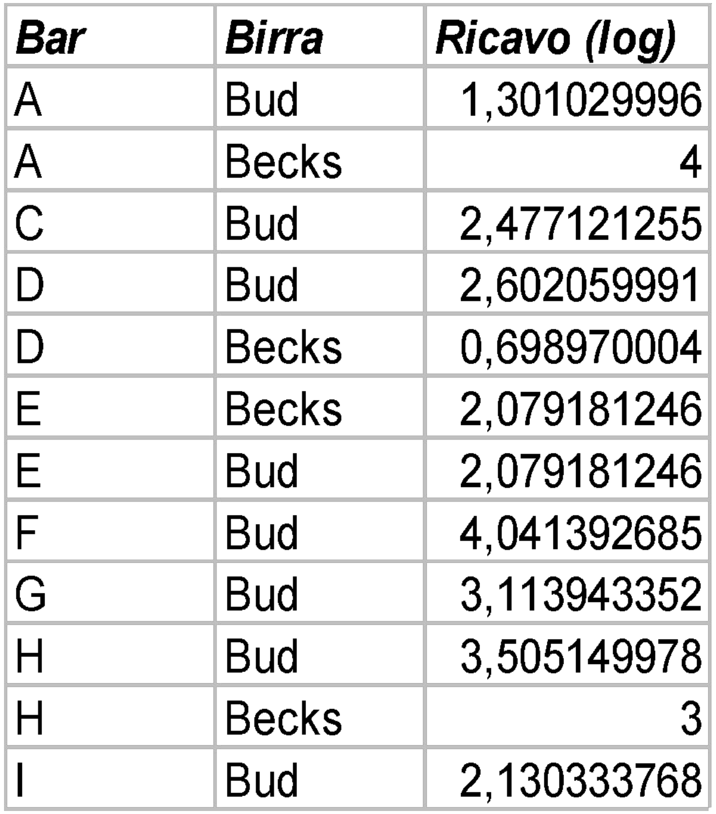


Bar	*Birra*	*Ricavo*
A	Bud	20
A	Becks	10000
C	Bud	300
D	Bud	400
D	Becks	5
E	Becks	120
E	Bud	120
F	Bud	11000
G	Bud	1300
H	Bud	3200
H	Becks	1000
I	Bud	135

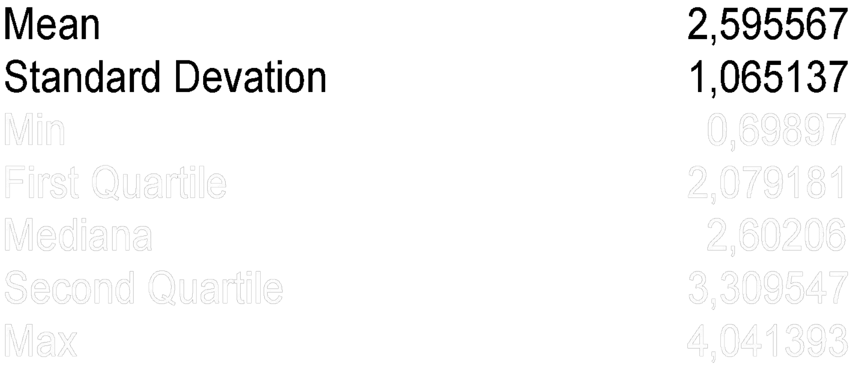
2481,8182	Mean
4079,0172	Standard Deviation
5	Min
120	1° Quartile
400	Median
2250	2° Quartile
11000	Max

Data are sparse!!!

Logarithmic Transformation: Example



Bar	*Birra*	*Ricavo (log)*
A	Bud	1,301029996
A	Becks	4
C	Bud	2,477121255
D	Bud	2,602059991
D	Becks	0,698970004
E	Becks	2,079181246
E	Bud	2,079181246
F	Bud	4,041392685
G	Bud	3,113943352
H	Bud	3,505149978
H	Becks	3
I	Bud	2,130333768



Mean	2,595567
Standard Devation	1,065137
Min	0,69897
First Quartile	2,079181
Mediana	2,60206
Second Quartile	3,309547
Max	4,041393

Stabilizing the Variance

$$T(x) = ax^p + b$$

- **Square-root Transformation**
- $p = 1/c$, c integer number
  - To make homogenous the variance of particular distributions e.g., Poisson Distribution
- **Reciprocal Transformation**
  - $p < 0$
  - Suitable for analyzing time series, when the variance increases too much wrt the mean

Discretization: Advantages

- Hard to understand the optimal discretization
    - We should need the real data distribution

- Original values can be **continuous** and **sparse**

- Discretized data can be **simple** to be interpreted

- Data distribution after discretization can have a **Normal shape**

- Discretized data can be too much **sparse yet**
    - Elimination of the attribute

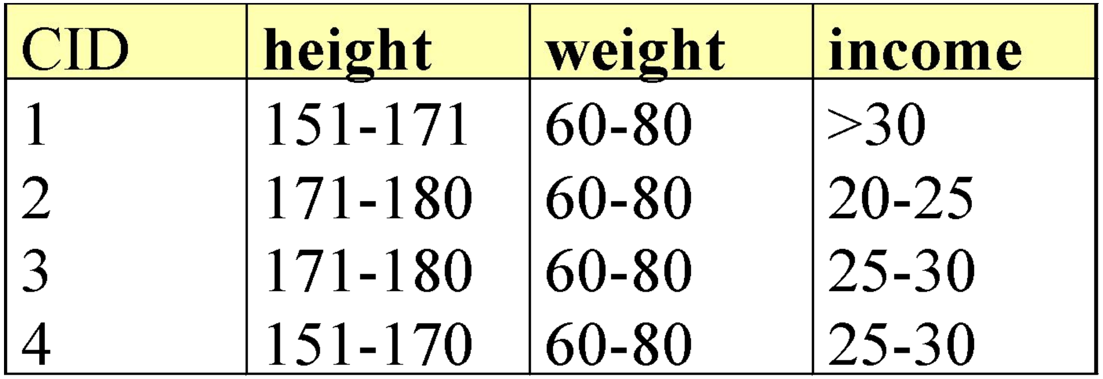
Unsupervised Discretization

- Characteristics:
  - No label for the instances
  - The number of classes is unknown

- Techniques of *binning*:
  - **Natural binning** □ Intervals with the same width
  - **Equal Frequency binning** □ Intervals with the same frequency
  - **Statistical binning** □ Use statistical information (Mean, variance, Quartile)

Discretization of quantitative attributes

- **Solution**: each value is replaced by the interval to which it belongs.
  - **height**: 0-150cm, 151-170cm, 171-180cm, >180c
  - **weight**: 0-40kg, 41-60kg, 60-80kg, >80kg
  - **income**: 0-10ML, 11-20ML, 20-25ML, 25-30ML, >30ML

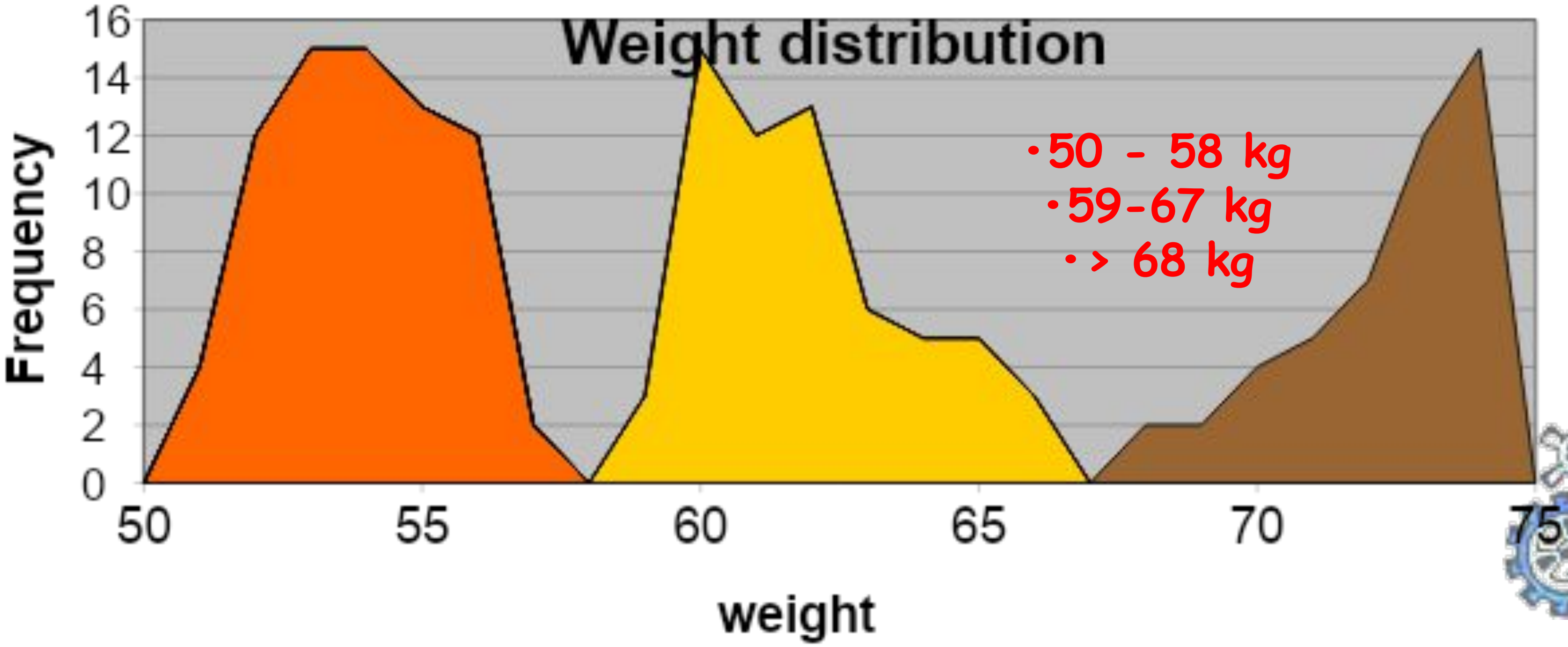


CID	**height**	**weight**	**income**
1	151-171	60-80	>30
2	171-180	60-80	20-25
3	171-180	60-80	25-30
4	151-170	60-80	25-30

- **Problem**: the discretization may be useless (see **weight**).

How to choose intervals?

1. **Interval with a fixed "reasonable" granularity**
   Ex. intervals of 10 cm for height.
2. **Interval size is defined by some domain dependent criterion**
   Ex.: 0-20ML, 21-22ML, 23-24ML, 25-26ML, >26ML
3. **Interval size determined by analyzing data, studying the distribution or using clustering**



- 50 - 58 kg
- 59-67 kg
- > 68 kg

Natural Binning

- Simple
- Sort of values, subdivision of the range of values in *k* parts with the same size

$$\delta = \frac{x_{\max} - x_{\min}}{k}$$

- Element x_j belongs to the class i if

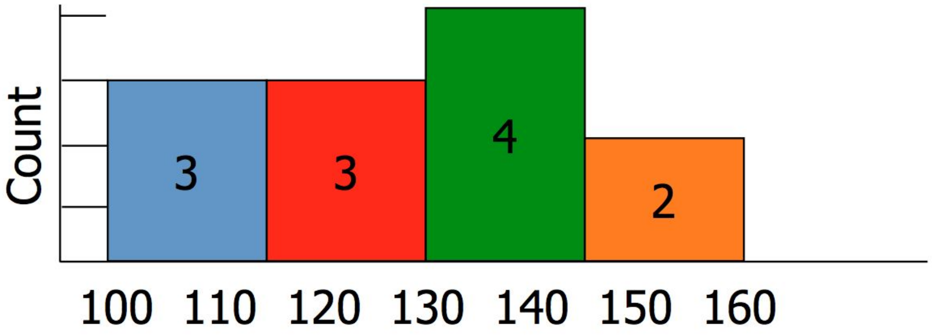
$$x_j \in [x_{min} + i\delta,\ x_{min} + (i+1)\delta)$$

- It can generate distribution very unbalanced

Example

Bar	Beer	Price
A	Bud	100
A	Becks	120
C	Bud	110
D	Bud	130
D	Becks	150
E	Becks	140
E	Bud	120
F	Bud	110
G	Bud	130
H	Bud	125
H	Becks	160
I	Bud	135

- δ =(160-100)/4 = 15
- class 1: [100,115)
- class 2: [115,130)
- class 3: [130,145)
- class 4: [145, 160]



Equal Frequency Binning

- Sort and count the elements, definition of k intervals of f, where:

$$f = \frac{N}{k}$$

(*N* = number of elements of the sample)

- The element x_i belongs to the class j if

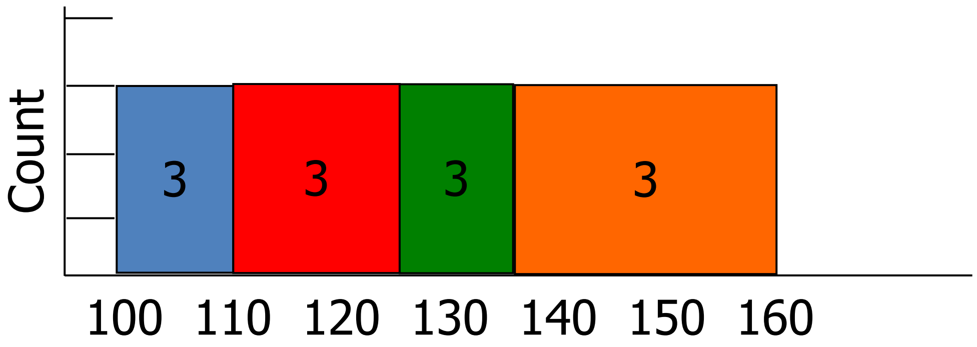
$$j \times f \leq i < (j+1) \times f$$

- It is not always suitable for highlighting interesting correlations

Example

Bar	Beer	Price
A	Bud	100
A	Becks	120
C	Bud	110
D	Bud	130
D	Becks	150
E	Becks	140
E	Bud	120
F	Bud	110
G	Bud	130
H	Bud	125
H	Becks	160
I	Bud	135

- $f = 12/4 = 3$
- class 1: {100,110,110}
- class 2: {120,120,125}
- class 3: {130,130,135}
- class 4: {140,150,160}



How many classes?

- If too few
  - ⇒ Loss of information on the distribution
- If too many
  - => Dispersion of values and does not show the form of distribution
- The optimal number of classes is function of N elements (Sturges, 1929)

$$C = 1 + \frac{10}{3}\log_{10}(N)$$

- The optimal width of the classes depends on the variance and the number of data (Scott, 1979)

$$h = \frac{3{,}5 \cdot s}{\sqrt{N}}$$

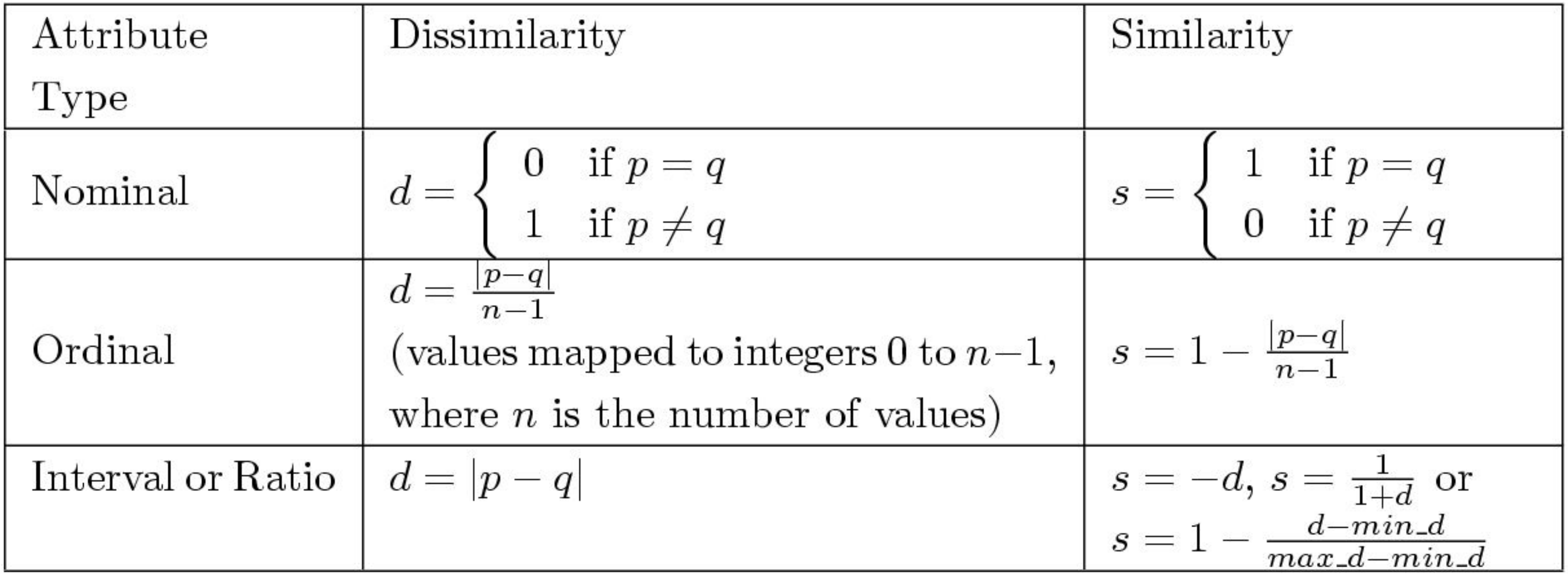
Similarity

Similarity and Dissimilarity

- **Similarity**
  - Numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]
- **Dissimilarity**
  - 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**

Similarity/Dissimilarity for ONE Attribute

p and *q* are the attribute values for two data objects.



Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p-q \rvert}{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{\lvert p-q \rvert}{n-1}$
Interval or Ratio	$d = \lvert p - q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Table 5.1. Similarity and dissimilarity for simple attributes

Many attributes: Euclidean Distance

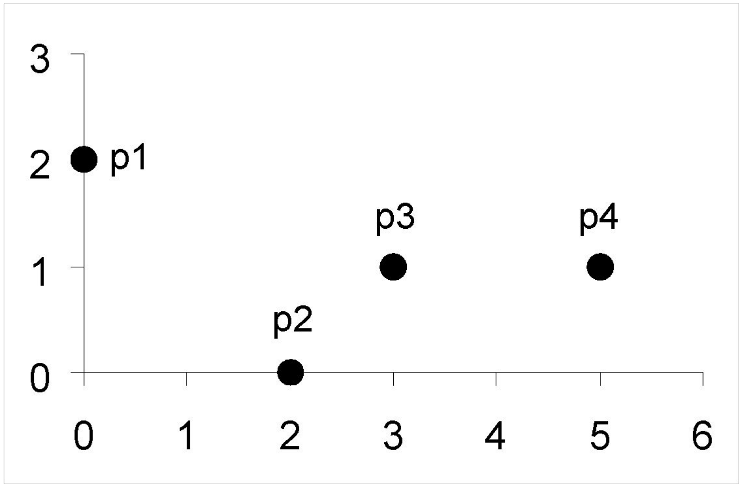
- Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

Where *n* is the number of dimensions (attributes) and p_k and q_k are, respectively, the value of kth attributes (components) or data objects *p* and *q*.

- Standardization is necessary, if scales differ.

Euclidean Distance



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

	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

Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

Where *r* is a parameter, *n* is the number of dimensions (attributes) and p_k and q_k are, respectively, the k_{th} attributes (components) or data objects *p* and *q*.

Minkowski Distance: Examples

- r = 1. City block (Manhattan, taxicab, L_1 norm) distance.
  - A common example of this is the Hamming distance, which is just the number of bits that are different between two binary vectors

- r = 2. Euclidean distance

- $r \rightarrow \infty$. "supremum" (L_{max} norm, L_∞ norm) distance.
  - This is the maximum difference between any component of the vectors

- Do not confuse r with n, i.e., all these distances are defined for all numbers of dimensions.

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

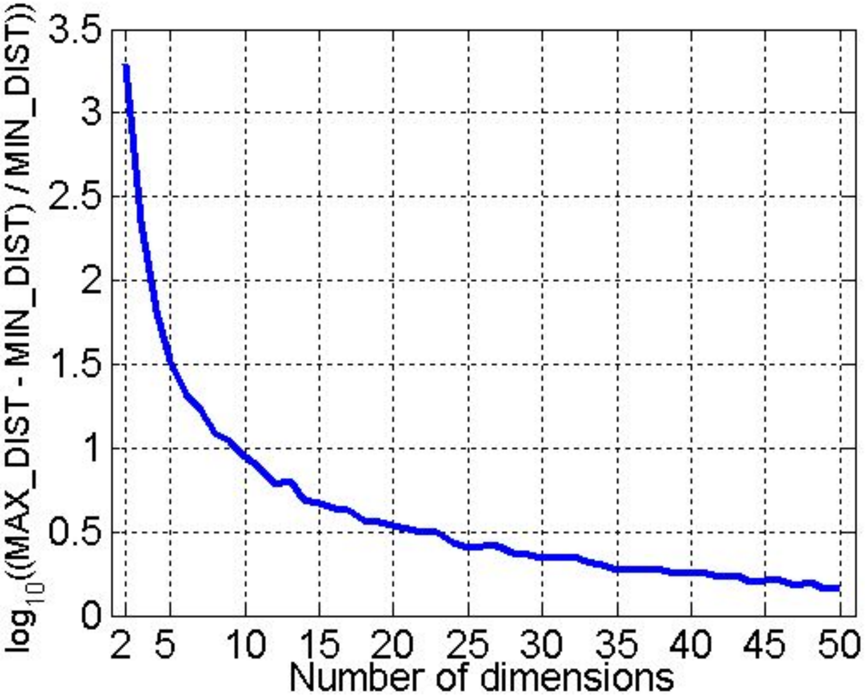
L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

Curse of Dimensionality

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies

- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful



- **Randomly generate 500 points**
- **Compute difference between max and min distance between any pair of points**

Common Properties of a Distance

- Distances, such as the Euclidean distance, have some well known properties.

1. $d(p, q) \geq 0$ for all p and q and $d(p, q) = 0$ only if $p = q$. (Positive definiteness)
2. $d(p, q) = d(q, p)$ for all p and q. (Symmetry)
3. $d(p, r) \leq d(p, q) + d(q, r)$ for all points p, q, and r. (Triangle Inequality)

where $d(p, q)$ is the distance (dissimilarity) between points (data objects), p and q.

- A distance that satisfies these properties is a metric

Common Properties of a Similarity

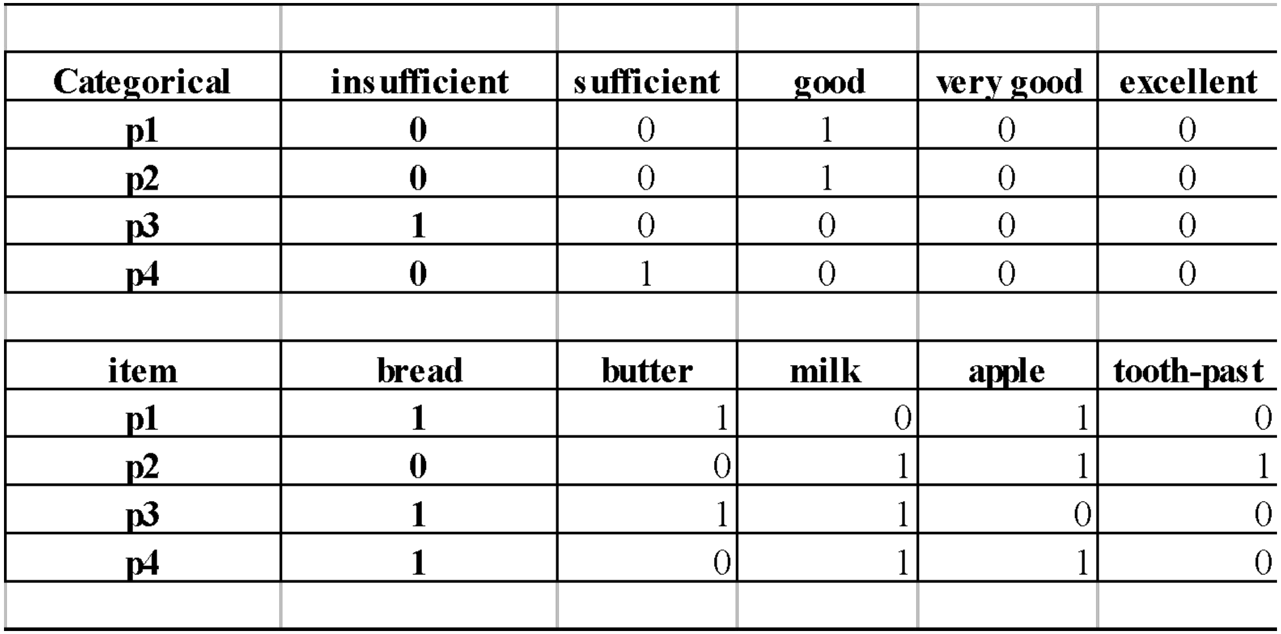
- Similarities, also have some well known properties.

1. $s(p, q) = 1$ (or maximum similarity) only if $p = q$.

2. $s(p, q) = s(q, p)$ for all p and q. (Symmetry)

where $s(p, q)$ is the similarity between points (data objects), p and q.

Binary Data



Categorical	insufficient	sufficient	good	very good	excellent
p1	0	0	1	0	0
p2	0	0	1	0	0
p3	1	0	0	0	0
p4	0	1	0	0	0

item	bread	butter	milk	apple	tooth-past
p1	1	1	0	1	0
p2	0	0	1	1	1
p3	1	1	1	0	0
p4	1	0	1	1	0

Similarity Between Binary Vectors

- Common situation is that objects, *p* and *q*, have only binary attributes
- Compute similarities using the following quantities

M_{01} = the number of attributes where p was 0 and q was 1
M_{10} = the number of attributes where p was 1 and q was 0
M_{00} = the number of attributes where p was 0 and q was 0
M_{11} = the number of attributes where p was 1 and q was 1

- Simple Matching and Jaccard Coefficients

SMC = number of matches / number of attributes
= $(M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

J = number of 11 matches / number of not-both-zero attributes values
= $(M_{11}) / (M_{01} + M_{10} + M_{11})$

SMC versus Jaccard: Example

p = 1 0 0 0 0 0 0 0 0 0

q = 0 0 0 0 0 0 1 0 0 1

$M_{01} = 2$ (the number of attributes where p was 0 and q was 1)
$M_{10} = 1$ (the number of attributes where p was 1 and q was 0)
$M_{00} = 7$ (the number of attributes where p was 0 and q was 0)
$M_{11} = 0$ (the number of attributes where p was 1 and q was 1)

SMC = $(M_{11} + M_{00})/(M_{01} + M_{10} + M_{11} + M_{00}) = (0+7) / (2+1+0+7) = 0.7$

J = $(M_{11}) / (M_{01} + M_{10} + M_{11}) = 0 / (2 + 1 + 0) = 0$

Document Data

	team	coach	play	ball	score	game	win	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

Cosine Similarity

- If d_1 and d_2 are two document vectors, then

  $$\cos(d_1, d_2) = (d_1 \cdot d_2) / ||d_1|| \ ||d_2|| ,$$

  where · indicates vector dot product and || *d* || is the length of vector *d*.

- 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**

  $d_1 \cdot d_2$= 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5
  $||d_1||$ = $(3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{\mathbf{0.5}}$ = $(42)^{\mathbf{0.5}}$ = 6.481
  $||d_2||$ = $(1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{\mathbf{0.5}}$ = $(6)^{\mathbf{0.5}}$ = 2.245

  $\cos(d_1, d_2)$ = .3150

Correlation

- Correlation measures the linear relationship between objects (binary or continuos)
- To compute correlation, we standardize data objects, p and q, and then take their dot product (covariance/standard deviation)

$$p'_k = (p_k - mean(p))$$

$$q'_k = (q_k - mean(q))$$

$$correlation(p,q) = (p' \bullet q') / (n-1) std(p) std(q)$$