# Data Mining

**Data mining means analyzing large data sets to identify and establish new patterns or relationships which in the end prove valid, comprehensible and useful.**

**Classical Statistics**
**Machine Learning**
**Pattern Recognition**

**Data Mining und Knowledge Discovery (DMKD) becomes part of AI (1990)**

***Data Mining:***
**Analysis, generation of hypotheses**

***Knowledge Discovery:***
**Evaluation and interpretation of hypotheses**

**Sources:** **Görz et al. (Eds.): Handbuch der künstlichen Intelligenz (3. Aufl.), Oldenbourg, 2000**
**Maimon & Rokach (Eds.): The Data Mining and Knowledge Discovery Handbook, Springer 2005**



# Data Mining Examples

**Example 1: Analysis of purchases in a supermarket**

| | | | | | |
|---|---|---|---|---|---|
| customer1 | pizza | beer | cheese | bread | chips |
| customer2 | milk | bread | ham | cigaretts | |
| customer3 | yoghurt | sugar | flour | cornflakes | napkins |
| customer4 | shampoo | beer | chips | newspaper | pizza |
| xustomer5 | chips | coffee | beer | pizza | cream |
| customer6 | jam | rolls | butter | beer | |
| ... | | | | | |

**If pizza and beer are in one purchase, it is likely that chips are also in that purchase.**

**Example 2: Monday is a likely day for error-prone production**

## Why is Data Mining a Problem?

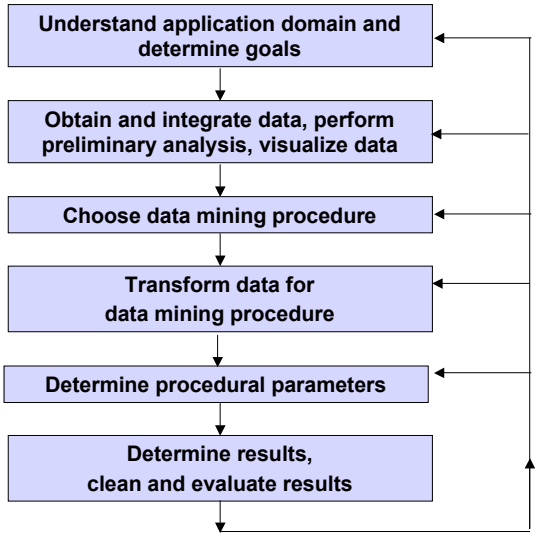
- Large data volume prohibits discovery of interesting relationships through human inspection or exhaustive search.
- It is difficult to discover something when you don´t know what to look for.
- It is easy to deceive yourself in discovering something which you expect.

***"There has always been a considerable number of people who busy themselves examining the last thousand numbers which have appeared on a roulette wheel, in search of some repeating pattern. Sadly enough, they have usually found it."***

***"The geographic density of stork nests is positively correlated with the birth rate of humans."***

3

## Structure of Data Mining and Knowledge Discovery




4

# Requirements for Association Rules

Determine rules X -> Y in transactions T where:

$$s(r) := \frac{\left|\left\{t \in T \mid X \cup Y \in t\right\}\right|}{|T|} \geq s_{min}$$ minimal support

$$c(r) := \frac{\left|\left\{t \in T \mid X \cup Y \in t\right\}\right|}{\left|\left\{t \in T \mid X \in t\right\}\right|} \geq c_{min}$$ minimal confidence

Typical values: $s_{min} = 0{,}01$ $c_{min} = 0{,}5$

{beer, pizza} -> {chips} will be established if
- at least 1% of all customers have bought beer, pizza, and chips, and
- at least 50% of the beer and pizza customers have also bought chips.

5

# Finding Association Rules (1)

Possible association rules may be ordered:

| | | | |
|---|---|---|---|
| a -> q | c = \|aq\| / \|a\| | s = \|aq\| / \|T\| | |
| ↓ | ?? | ≥ | refining a premise |
| ab -> q | c = \|abq\| / \|ab\| | s = \|abq\| / \|T\| | |

| | | | |
|---|---|---|---|
| a -> q | c = \|aq\| / \|a\| | s = \|aq\| / \|T\| | |
| ↓ | ≥ | ≥ | refining a consequent |
| a -> bq | c = \|abq\| / \|a\| | s = \|abq\| / \|T\| | |

- A rule search based on successive refinements can be pruned by prescribing confidence $c_{min}$ and support $s_{min}$.
- Pruning by support can be done without distinguishing between premise and consequence.

6

# Finding Association Rules (2)

Phase 1: Find frequent itemsets ($s \geq s_{min}$ )

Phase 2: Determine association rules in frequent itemsets ($c \geq c_{min}$ )

Efficient procedure for Phase 1:

To obtain frequent (k+1)-sets from k-sets, first consider all pairs of k-sets with (k-1) common items since both are frequent.

**Closure under Support:**
Given two itemsets X and Y with $X \subseteq Y$, then $s(X) \geq s(Y)$.

Efficient procedure for Phase 2:

Itemsets with sufficient support are transformed into rules by beginning with 1-item consequences and shifting additional items from the premise to the consequence until the confidence falls below the threshold.

| | | |
|---|---|---|
| ab -> c | c = \|abc\| / \|ab\| | s = \|abc\| / \|T\| |
| ↓ | ≥ | = |
| a -> bc | c = \|abc\| / \|a\| | s = \|abc\| / \|T\| |

7

# APRIORI Algorithm

```
proc APRIORI(I, T, s_min, c_min)
    L := FREQUENT-SETS(I, T, smin)
    R := RULES(L, cmin)
    return R
```

```
proc FREQUENT-SETS(I, T, s_min)
    C_1 := i∈I { {i} }
    L_1 := PRUNE(C_1)
    while L_k ≠ ∅
        C_k+1 := CANDIDATES(L_k)
        L_k+1 := PRUNE(C_k+1, T)
        k := k+1
    return j=2..k{ L_j }
```

```
proc RULES(L, c_min)
    R := ∅
    forall l ∈ L, k := |l| ≥ 2
        H_1 := i∈l {{i}}, m := 1
        loop
            forall h ∈ H_m
            if s(l_k) / s(l_k \h) ≥ c_min
            then add l_k \h → h to R
            else H_m := H_m \{h}
        while m ≤ k-2
        H_m+1 := CANDIDATES(H_m)
        m := m+1
    return R
```

8

# Example for APRIORI Algorithm

**T: abcd abf abce abch aci bci cgh de** $c_{min} = 0.8$ $s_{min} = 3/8$

**FREQUENT-SETS:**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_1$: | a | b | c | d | e | f | g | h | i | $C_2$: | ab | ac | bc | $C_3$: | abc |
| s: | 5 | 5 | 6 | 2 | 2 | 1 | 1 | 2 | 2 | s: | 4 | 4 | 4 | s: | 3 |
| $L_1$: | a | b | c | | | | | | | $L_2$: | ab | ac | bc | $L_3$: | abc |

**RULES:**

**ab:** $H_1$ = ab
**c(a->b) = s(ab)/s(a) = 4/5 = 0.8**
**c(b->a) = s(ab)/s(b) = 4/5 = 0.8**

**ac:** $H_1$ = ac
**c(a->c) = s(ac)/s(a) = 4/5 = 0.8**
c(c->a) = s(ac)/s(c) = 4/6 = 0.66

**bc:** $H_1$ = bc
**c(b->c) = s(bc)/s(b) = 4/5 = 0.8**
c(c->b) = s(bc)/s(c) = 4/6 = 0.66

**abc:** $H_1$ = abc
c(ab->c) = s(abc)/s(ab) = 3/4 = 0.75
c(ac->b) = s(abc)/s(ac) = 3/4 = 0.75
c(bc->a) = s(abc)/s(bc) = 3/4 = 0.75

$H_2 = \varnothing$

9

# Limits of Support and Confidence

**A rule X -> Y expresses a causality which may not be justified even if $s \geq s_{min}$ and $c \geq c_{min}$.**

**Example:**

**$s_{min}$ = 0,01 $c_{min}$ = 0,5**
**70% of all customers buy bread**
**2% of all customers buy soap (independently of bread)**
**s(soap -> bread) ≈ 0,02*0,7 = 0,014**
**c(soap -> bread) ≈ 0,02*0,7/0,02 = 0,7**

**Customers who buy soap typically also buy bread (???)**

**To avoid meaningless rules, postprocessing is required.**
**Heuristics for discarding X -> Y:**
**P(XY) ≈ P(X)*P(Y)** premise and consequence are statistically independent
**$P(Y) \geq c_{min}$** influence of premise on consequence is negligible

**But support and confidence remain essential for the derivation process.**

10

# Measures of Interestingness

To select interesting rules from those with sufficient confidence and support, several measures of interestingness have been proposed.

The Kullbach-Leibler Divergence (KLD) measures the "distance" between a distribution $p(x_i)$ and its approximation $q(x_i)$ in terms of entropy:

$$KLD(p,q) = \sum_{i=1..N} p(x_i) \log_2 \frac{1}{q(x_i)} - \sum_{i=1..N} p(x_i) \log_2 \frac{1}{p(x_i)}$$

$$= \sum_{i=1..N} p(x_i) \log_2 \frac{p(x_i)}{q(x_i)}$$

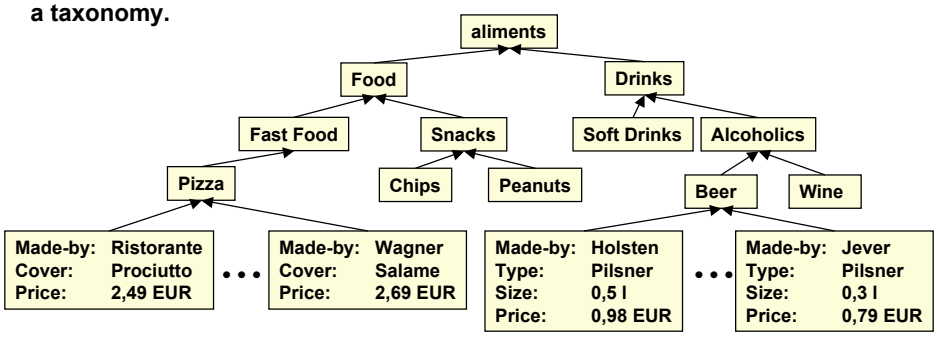
For a rule A -> B, KLD can measure the information gain of B from A by comparing the distributions p(y|x=A) with p(y) where $y \in \{B, \neg B\}$.

$$KLD(p(y \mid x = A), p(y)) = \sum_{y \in \{B, \neg B\}} p(y \mid x = A) \log_2 \frac{p(y \mid x = A)}{p(y)}$$

11

# Data Mining of Structured Data

Instead of unstructured symbols, items may be structured and embedded in a taxonomy.



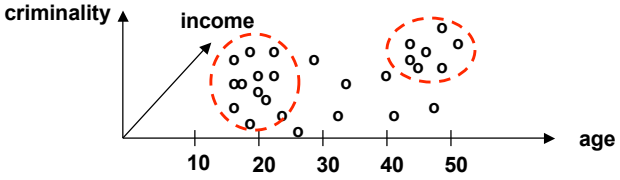

- Itemsets may be composed of items from different levels, but a more general item in an itemset precludes the presence of its specializations.
- Support at higher levels may be computed from support at lower levels.
- A domain-dependent measure of interestingness may be required to select useful itemsets and rules.

12

# Discovering Clusters in Data

***"Understanding our world requires conceptualizing the similarities and differences between the entities that compose it."*** **Tyron & Bailey 1970**

**"Clusters" of data objects are hypothetical classes based on similarities and distances. Data objects should be as similar as possible within clusters and as distinct as possible between clusters.**




**Cluster 1: age 15 - 25, low income, high criminality ("youth criminality")**
**Cluster 2: age 45 - 55, high income, high criminality ("white-collar criminality")**

**Data objects are viewed as points in a multi-dimensional feature space. Similarity of data is judged by distance measures.**

13

# Main Problems of Clustering

- **Determine useful features of data objects**
  ***Is e.g. body size a useful feature for social clustering?***
- **Collect representative data**
  ***Clusters from statitically biased samples may be misleading***
- **Determine similarity measure**
  ***What is the "distance" between e.g. male and female? How does it compare to e.g. age or income distances?***
- **Determine granularity or cluster number**
- **Determine clusters**

14

## Distance Measures

A valid distance measure between two data objects $x_i$ and $x_j$ must satisfy

- $d(x_i, x_j) \geq 0$
- $d(x_i, x_j) = d(x_j, x_i)$
- $x_i = x_j \Rightarrow d(x_i, x_j) = 0$

A distance measure is a metric if

- $d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k)$
- $d(x_i, x_j) = 0 \Rightarrow x_i = x_j$

Distance measures depend on the data types of the featureswhich must be compared:

- Continuous-valued
- Discrete-valued
- Binary-valued
- Symbol-valued
- Ordinal-valued

15

## Distance for Numeric Features (1)

$x_i$ and $x_j$ are continuous-valued N-dimensional data objects

Weighted distance: $d(x_i, x_j) = w_1 |x_{i1}-x_{j1}|^g + w_2 |x_{i2}-x_{j2}|^g + \ldots w_N |x_{iN}-x_{jN}|^g$

For $w_1 \ldots w_N = 1$ we have

- g = 1: Manhattan metric
- g = 2: Euclidean metric
- g -> ∞: Chebychev metric
  (emphasizes the dimension with largest distance)

$x_i$ and $x_j$ are discrete-valued N-dimensional data objects

Example: number-of-children {0, 1, 2, ... }

All distance measures for continuous-valued features can be in principle applied.

Sometimes distances at large values are less important than at small small values:

$$d(x_{ik}, x_{jk}) = \frac{|x_{ik} - x_{jk}|}{x_{ik} + x_{jk}} \quad \text{for } x > 0$$

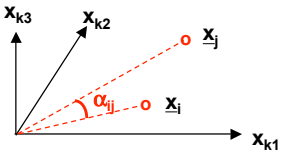
16

# Distance for Numeric Features (2)

$x_i$ and $x_j$ are continuous-valued N-dimensional data objects, viewed as vectors $\underline{x}_i$ and $\underline{x}_j$

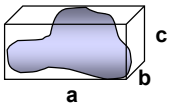
If the angle between $\underline{x}_i$ and $\underline{x}_j$ is significant for the distance, rather than the magnitude of each vector, the cosine distance is appropriate:

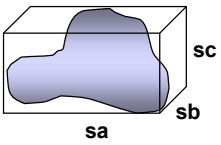
$$d(\underline{x}_i, \underline{x}_j) = 1 - \frac{\underline{x}_i^T \underline{x}_j}{\|x_i\| \|x_j\|} = 1 - \cos\alpha_{ij}$$




Example:

Scale invariant decription of shape by bounding box [x, y, z]

$\underline{x}_1$: 


$\underline{x}_2$: 


$d(\underline{x}_1, \underline{x}_2) = 0$

17

# Distance for Binary Features

$x_i$ and $x_j$ are binary-valued N-dimensional data objects

Distance is determined by counting equal and unequal 0 and 1 features.

For symmetric binary features, 0 and 1 are equally valued:

$$d(x_i, x_j) = \frac{|\text{unequal features}|}{N}$$ Hamming distance

For asymmetric binary features, 0 is often considered less valued, and features where both objects are 0 are ignored:

q = |equal 0,0 features| r = |equal 1,1 features|

s = |unequal 0,1 features| t = |unequal 1,0 features|

$$d(x_i, x_j) = \frac{s + t}{r + s + t}$$

18

## Distance for Nominal Features

$x_i$ and $x_j$ are symbol-valued N-dimensional data objects

Examples: colour ∈ {red, green, blue, black, white}
sex ∈ {male, female}

1. Matching distance $d(x_i, x_j) = \frac{N - m}{N}$ m = number of matches

2. Transformation to binary features

A k-valued nominal feature is transformed into k binary features.

Example: colour ∈ {red, green, blue, black, white}

red ∈ {T, F}, green ∈ {T, F}, blue ∈ {T, F}, ...

After transformation, distance measures for bianry features can be applied.

19

## Distance Metrics for Mixed-typed Features

Distances between features of different types can be combined by first normalizing the typed distance measures to the range [0 .. 1] and then using a distance measure for numeric values.

Normalization of continuous-valued feature distance:

$$d(x_{ik}, x_{jk}) = \left( \frac{| x_{ik} - x_{jk} |}{\max x_k - \min x_k} \right)^g$$

The problem of combining "apples with pears" cannot be solved satisfactorily, not only for different data types but generally for features belonging to different semantic categories.

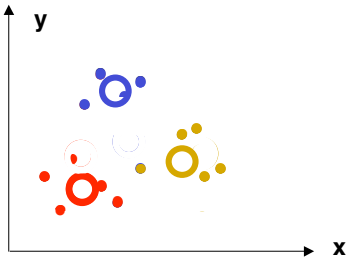
20

# K-means Clustering

- Most popular clustering algorithm
- Searches for local minimum of sum of Euclidean sample distances to cluster centers
- Guaranteed convergence in a finite number of steps
- Requires initialization of fixed number of clusters k
- May converge to local minimum

| | |
|---|---|
| A | Initialize cluster centers |
| B | Assign data objects to nearest cluster centers |
| C | New cluster centers are the mean of assigned data objects |
| D | Repeat steps B to D until no more changes occur |

<u>Example</u>:
K-means clustering with k = 3






# EM-Algorithm

K-means clustering is a special case of the Expectation-Maximization (EM) algorithm.

<u>Basic idea of EM-algorithm</u>:

Initially one has
- data with missing values, e.g. unassigned cluster memberships
- distribution models, e.g. rule to assign to nearest-distance cluster means

Iterate the two steps:

E-Step: Compute expected distribution parameters based on data (initially by random choice)

M-Step: Maximize likelihood of missing values based on distribution parameters

# Example for Expectation Maximization

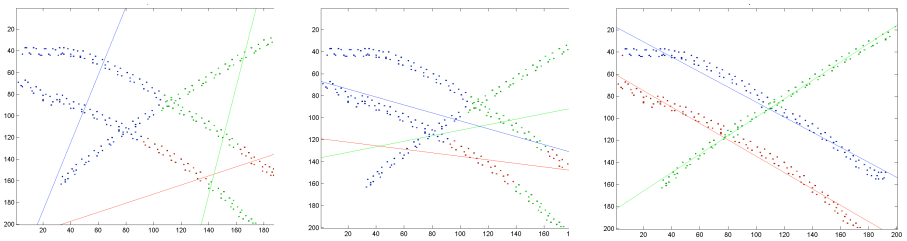
**Consider the problem of fitting 3 straight lines to data, not knowing which data belong to which line.**

**(Example by Anna Ergorova, FU Berlin)**

**Algorithm:**

| | |
|---|---|
| **A** | **Select 3 random lines initially** |
| **B** | **Assign data points to each line by minimum distance criterion** |
| **C** | **Determine best-fitting straight line for assigned data points** |
| **D** | **Repeat B to D until no further changes occur** |

**1. Iteration (after B)** **1. Iteration (after C)** **6. Iteration (after B)**



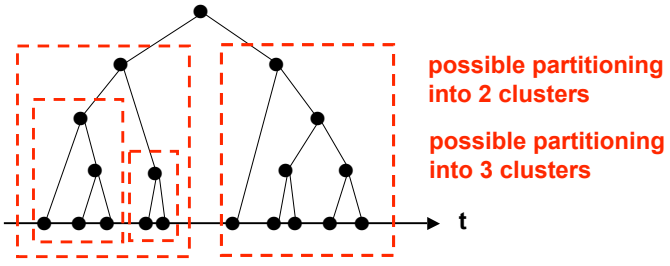
23

# Agglomerative Clustering

- **Data are incrementally combined to clusters**
- **A cluster tree is generated**
- **Final partitioning into clusters is left to the user**

| | |
|---|---|
| **A** | **Initially, all data objects are distinct clusters** |
| **B** | **Merge cluster pair with nearest distance** |
| **C** | **Enter new cluster into cluster tree** |
| **D** | **Repeat steps B to D until all clusters are merged** |

**Example:**
**Clustering of 1-dimensional data objects**




24

# Distance Measures for Clusters

Intra-cluster distance can be measured by

- the average distance
- the maximum distance

between cluster members and cluster center.

(Refer to the distance measures introduced earlier)

Inter-cluster distance can be measured by

- the smallest distance between elements of two distinct clusters ("single-link clustering")
- the largest distance between elements of two distinct clusters ("complete-link clustering")
- the average distance between elements of two distinct clusters ("average-link clustering")

What are the effects of the different inter-cluster distance measures on the results of agglomerative clustering?

25