Ludwig-Maximilians-Universität München
Lehrstuhl für Datenbanksysteme und Data Mining
Prof. Dr. Thomas Seidl

Knowledge Discovery and Data Mining 1

(Data Mining Algorithms 1)

Winter Semester 2019/20

Agenda

People

Lecturer

- Prof. Dr. Thomas Seidl

Assistants

- Maximilian Huenemoerder
- Janina Sontheim

Student Assistants

- Marcel Baur
- Florian von Keller

Schedule

Lecture (begins: 15.10.2018)

Tue	09:15-11:45	E004 (HGB)

Tutorials (begin: 24.10.2018)

Thu	12:15-13:45	Lehrturm-VU107 (Prof.-Huber-Pl. 2)
Thu	14:15-15:45	Lehrturm-VU107 (Prof.-Huber-Pl. 2)
Fri	12:15-13:45	Lehrturm-V005 (Prof.-Huber-Pl. 2)
Fri	14:15-15:45	Lehrturm-V005 (Prof.-Huber-Pl. 2)

Exam

1. Hauptklausur: tba
2. Nachholklausur: tba

Material, Tutorials & Exam

Material (Slides, Exercises, etc.)

Available on course webpage:

http://www.dbs.ifi.lmu.de/cms/studium_lehre/lehre_master/kdd1920/index.html

Tutorial

- Python Introduction now available on website
- New exercise sheets available weekly
- Prepare at home
- Presentation and discussion one week after

Exam

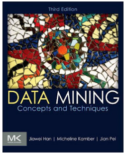
- Written exam at the end of semester
- All material discussed in the lecture and tutorials
- Registration via Uni2Work

Agenda

Textbook / Acknowledgements

The slides used in this course are modified versions of the copyrighted original slides provided by the authors of the adopted textbooks:

- © Jiawei Han, Micheline Kamber, Jian Pei: *Data Mining – Concepts and Techniques*, 3rd ed., Morgan Kaufmann Publishers, 2011. http://www.cs.uiuc.edu/~hanj/bk3



- © Martin Ester and Jörg Sander: *Knowledge Discovery in Databases – Techniken und Anwendungen* Springer Verlag, 2000 (in German).

Content of the Course

- Introduction
- Preliminaries – what is data, how to represent data, how to present data
- Classification – supervised learning
- Clustering – unsupervised learning
- Frequent Pattern Mining – itemsets, sequences, processes
- Further topics – outlook

Motivation

- Data Mining = extraction of patterns from data
- Patterns
  - Regularities – examples: frequent itemsets, clusters
  - Irregularities – examples: outliers
- Not all patterns are useful
  - *"all mothers in our database are female"* $\rightsquigarrow$ trivial/known
  - *"bread, butter is frequent"* given *"bread, butter, salt is frequent"* $\rightsquigarrow$ redundant
- Aggregation of data may help: Basic statistics

What is Data Mining?

Knowledge Discovery in Databases (Data Mining)

Extraction of interesting *(non-trivial, implicit, previously unknown and potentially useful)* information or patterns from data in *large databases*

Roots of Data Mining

- Statistics
- Machine Learning
- Database Systems
- Information Visualization

Data Mining and Machine Learning

Descriptive Learning

- Better understanding – data mining
- examples: pattern recognition, clustering, outlier detection

Predictive Learning

- Better forecasts – regression
- examples: traffic prediction, labeling, fraud detection

Prescriptive Learning

- Better actions – artificial intelligence
- examples: predictive maintenance, autonomous driving, medical therapies

Data Mining: Motivation

"Necessity is the mother of invention"

Data Explosion Problem

Tremendous amounts of data caused by

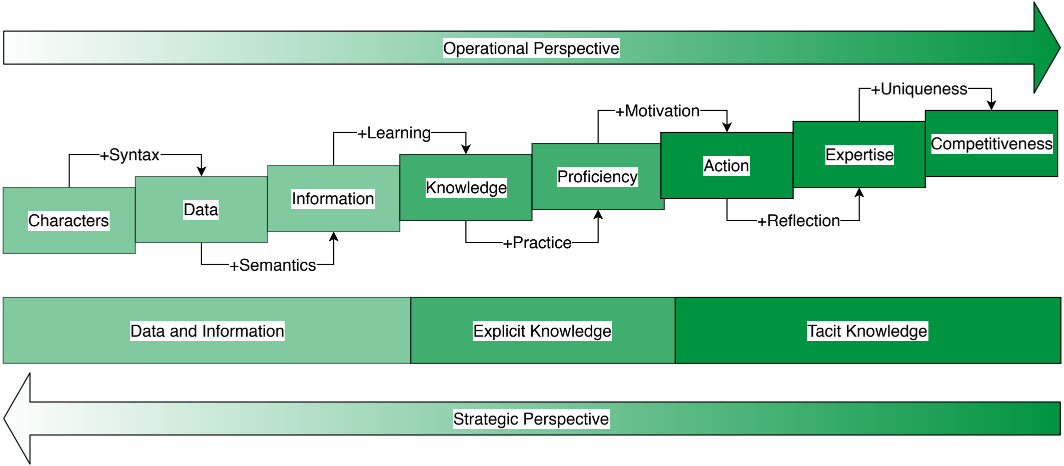
- Automated data collection
- Mature database technology

"We are drowning in data, but starving for knowledge!"

Solution

- Data Warehousing and on-line analytical processing (OLAP)
- Data Mining: Extraction of interesting knowledge (rules, regularities, patterns, constraints) from data in large databases

Data Mining: Motivation



Stairs of Knowledge (K. North) [1]

[1] Stairs of Knowledge: North, K.: Wissensorientierte Unternehmensführung - Wertschöpfung durch Wissen. Gabler, Wiesbaden 1998.

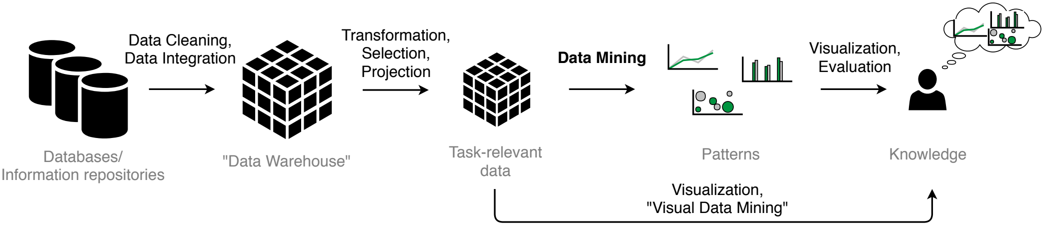
Data Mining: Potential Applications

- Database analysis and decision support
  - *Market analysis and management*:
    target marketing, customer relation management, market basket analysis, cross selling, market segmentation
  - *Risk analysis and management*:
    Forecasting, customer retention ("Kundenbindung"), improved underwriting, quality control, competitive analysis
  - *Fraud detection and management*
- Other Applications:
  - Text mining (news group, email, documents) and Web analysis.
  - Intelligent query answering

Agenda

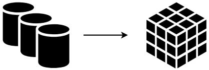
The Knowledge Discovery Process

- The KDD-Process (Knowledge Discovery in Databases)



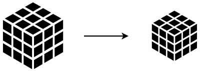
- Data Mining:
  - Frequent Pattern Mining
  - Clustering
  - Classification
  - Regression
  - Process Mining
  - ...

KDD Process: Data Cleaning & Integration



- . . . may take 60% of effort
- Integration of data from different sources
  - Mapping of attribute names, e.g. `C_Nr` → `O_Id`
  - Joining different tables, e.g. `Table1 = [C_Nr, Info1]` and `Table2 = [O_Id, Info2]`
    ⇝ `JoinedTable = [O_Id, Info1, Info2]`
- Elimination of inconsistencies
- Elimination of noise
- Computation of missing values (if necessary and possible): Possible strategies e.g. default value, average value, or application specific computations

KDD Process: Focusing on Task-Relevant Data



Task

- Find useful features, dimensionality/variable reduction, invariant representation
- Creating a target data set

Selections

Select the relevant tuples/rows from the database tables, e.g., sales data for the last year

KDD Process: Focusing on Task-Relevant Data

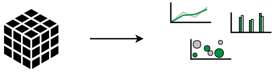
Projections

Select the relevant attributes/columns from the database tables, e.g., (`id`, `name`, `date`, `location`, `amount`) $\rightsquigarrow$ (`id`, `date`, `amount`)

Transformations, e.g.:

- Discretization of numerical attributes, e.g.,
  `amount`: [0, 100] $\rightsquigarrow$ `d_amount`: {low, medium, high}
- Computation of derived tuples/rows and derived attributes:
  - aggregation of sets of tuples, e.g., total amount per months
  - new attributes, e.g., `diff = sales current month - sales previous month`

KDD Process: Basic Data Mining Tasks



Goal

Find patterns of interest

Tasks

- Identify task: Are there labels (in the training data)?
  - *Many* $\rightsquigarrow$ Supervised learning (focus on given concepts)
  - *Some few* $\rightsquigarrow$ Semi-supervised learning (focus on few hidden concepts)
  - *None* $\rightsquigarrow$ Unsupervised learning (many hidden concepts)
- Choose fitting mining algorithm(s)

Basic Mining Tasks: Frequent Itemset Mining

Setting

Given a database of transactions, e.g.

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Motivation

Frequently co-occurring items in the set of transactions indicate correlations or causalities

Examples

- `buys(x, "diapers")` $\Rightarrow$ `buys(x, "beers")` [supp: 0.5%, conf: 60%]
- `major(x, "CS")` $\wedge$ `takes(x, "DB")` $\Rightarrow$ `grade(x,"A")` [supp: 1.0%, conf: 75%]

Basic Mining Tasks: Frequent Itemset Mining

Applications

- Market-basket analysis
- Cross-marketing
- Catalogue design
- Also used as a basis for clustering, classification
- Association rule mining: Determine correlations between different itemsets

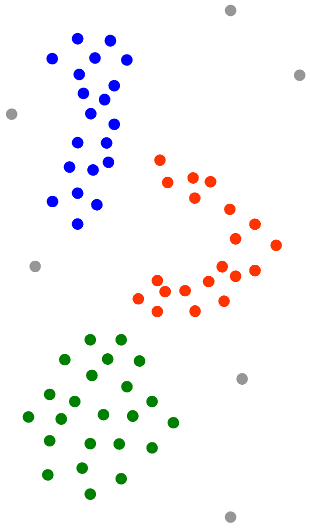
Basic Mining Tasks: Clustering

Setting

- Database of objects O
- Unknown class labels
- Similarity model for objects, often as (dis)similarity function $sim : O \times O \to \mathbb{R}$

Task

Group objects into clusters while maximizing intra-cluster similarity (cohesion) and minimizing inter-cluster similarity (separation)



Basic Mining Tasks: Clustering

Applications

- Customer profiling/segmentation
- Document or image collections
- Web access patterns
- ...

Basic Mining Tasks: Classification

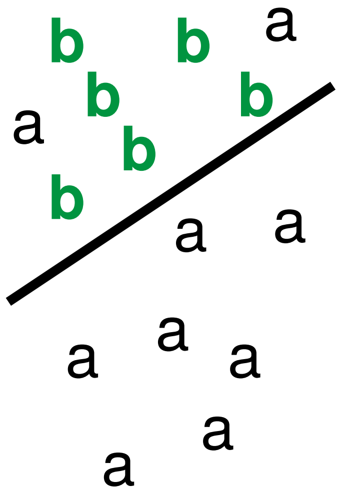
Setting

Class labels are known for a small set of "training data"

Task

Find models/functions/rules (based on attribute values of the training examples) that

- describe and distinguish classes
- predict class membership for “new” objects



Basic Mining Tasks: Classification

Applications

- Classify disease type for tissue samples from gene expression values
- Automatic assignment of categories to large sets of newly observed celestial objects
- Predict unknown or missing values (cf. KDD data cleaning & integration)
- ...

Basic Mining Tasks: Regression

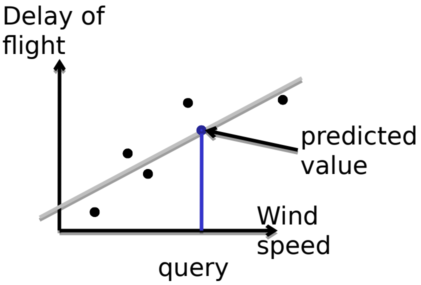
Setting

Numerical output values are known for a small set of "training data"

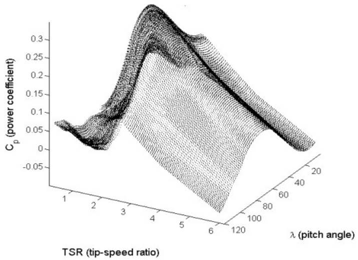
Task

Find models/functions/rules (based on attribute values of the training examples) that

- describe the numerical output values of the training data
- predict the numerical value for "new" objects



Basic Mining Tasks: Regression

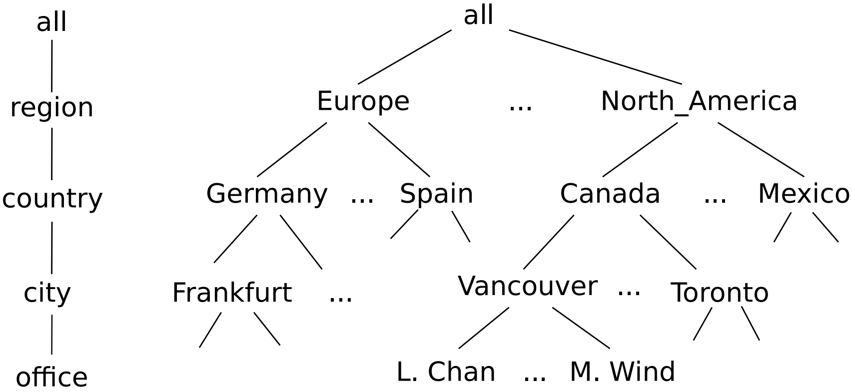


Applications

- Build a model of the housing values, which can be used to predict the price for a house in a certain area
- Build a model of an engineering process as a basis to control a technical system
- ...

Basic Mining Tasks: Generalization Levels

- Generalize, summarize, and contrast data characteristics
- Based on attribute aggregation along concept hierarchies
  - Data cube approach (OLAP)
  - Attribute-oriented induction approach



Basic Mining Tasks: Other Methods

Outlier Detection

Find objects that do not comply with the general behaviour of the data (fraud detection, rare events analysis)

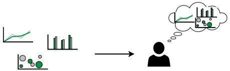
Trends and Evolution Analysis

Sequential patterns (find re-occurring sequences of events)

Methods for special data types, and applications

- Process Mining
- Spatial Data Mining
- Graphs
- ...

KDD Process: Evaluation and Visualization



- Pattern evaluation and knowledge presentation: Visualization, transformation, removing redundant patterns, etc.
- Different stages of visualization:
  - visualization of *data*
  - visualization of data mining *results*
  - visualization of data mining *processes*
  - *interactive* visual data mining
- Different types of 2D/3D plots, charts and diagrams are used, e.g. box-plots, trees, scatterplots, parallel coordinates
- Supports insights and usage of discovered knowledge

Summary

- Data mining = Discovering interesting patterns from large amounts of data
- A natural evolution of database technology, machine learning, statistics, visualization, in great demand, with wide applications
- A KDD process includes data cleaning, data integration, data selection, transformation, data mining, pattern evaluation, and knowledge presentation
- Data mining functionalities: characterization, discrimination, association, classification, clustering, outlier and trend analysis, etc.

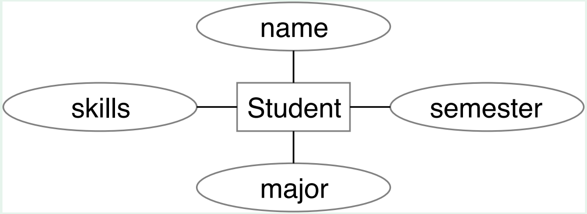
References: Where to find scientific publications

	Conference	Journal
Data Mining and KDD	KDD, PKDD, SDM, PAKDD, ICDM, ...	Data Mining and Knowledge Discovery, ...
Database Field	ACM-SIGMOD, ACM-PODS, VLDB, ICDE, EDBT, CIKM, ...	ACM-TODS, J. ACM, IEEE-TKDE, JIIS, VLDBJ, ...
AI and Machine Learning	Machine learning, AAAI, IJCAI, ICLR, ...	Machine Learning, Artificial Intelligence, ...
Statistics	Joint Stat. Meeting, ...	Annals of Statistics, ...
Visualization	CHI (Comp. Human Interaction), ...	IEEE Trans. Visualization and Computer Graphics, ...

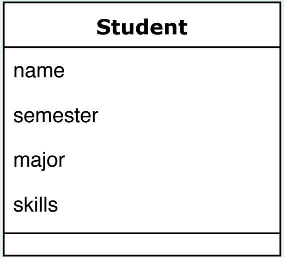
Agenda

Objects and Attributes (Conceptual Modeling)

Entity-Relationship Diagram (ER)



UML Class Diagram



Data Tables (Relational Model)

name	sem	major	skills
Ann	3	CS	Java, C, R
Bob	1	CS	Java, PHP
Charly	4	History	Piano
Debra	2	Arts	Painting

Overview of (Attribute) Data Types

Simple Data Types

Numeric/metric, Categorical/nominal, ordinal

Composed Data Types

Sets, sequences, vectors

Complex Data Types

- Multimedia: images, videos, audio, text, documents, web pages, etc.
- Spatial, geometric: shapes, molecules, geography, etc.
- Structures: graphs, networks, trees, etc.

Simple Data Types: Numeric Data

Numeric Data

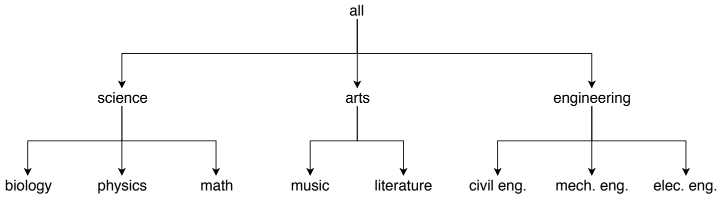
- Numbers: natural, integer, rational, real numbers
- Examples: age, income, shoe size, height, weight
- Comparison: difference (absolute value)
- Example: 3 is more similar to 30 than to 3,000

Simple Data Types: Categorical Data

- "Just identifiers"
- Examples:
  - occupation = { butcher, hairdresser, physicist, physician, ... }
  - subjects = { physics, biology, math, music, literature, ... }
- Comparison: How to compare values?
  - Trivial metric:

$$d(p,q) = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{else} \end{cases}$$

  - Generalization hierarchy: Use path length



  - Dedicated similarity matrix, $O \times O \to \mathbb{R}_0^+$

Generalization: Metric Data

Metric Space

Metric space (O, d) consists of object set O and *metric distance* function $d : O \times O \to \mathbb{R}_0^+$ which fulfills:

$$\begin{aligned} \text{Symmetry:} \quad & \forall p, q \in O : d(p, q) = d(q, p) \\ \text{Identity of Indiscernibles:} \quad & \forall p, q \in O : d(p, q) = 0 \iff p = q \\ \text{Triangle Inequality:} \quad & \forall p, q, o \in O : d(p, q) \leq d(p, o) + d(o, q) \end{aligned}$$

Example: Points in 2D space or in $\mathbb{R}^n$ with Euclidean distance

Simple Data Types: Ordinal

Characteristic

There is a (total) order $\leq$ on the set of possible data values O:

$$
\begin{aligned}
\text{Transitivity:} \quad & \forall p, q, o \in O : p \leq q \land q \leq o \implies p \leq o \\
\text{Antisymmetry:} \quad & \forall p, q \in O : p \leq q \land q \leq p \implies p = q \\
\text{Totality:} \quad & \forall p, q \in O : p \leq q \lor q \leq p
\end{aligned}
$$

Examples

- Words & lexicographic ordering: *high* $\leq$ *highschool* $\leq$ *highscore*
- (Vague) sizes: *tiny* $\leq$ *small* $\leq$ *medium* $\leq$ *big* $\leq$ *huge*
- Frequencies, e.g.: *never* $\leq$ *seldom* $\leq$ *rarely* $\leq$ *occasionally* $\leq$ *sometimes* $\leq$ *often* $\leq$ *frequently* $\leq$ *regularly* $\leq$ *usually* $\leq$ *always*

Composed Data Types: Sets

Characteristic

Unordered collection of individual values

Example

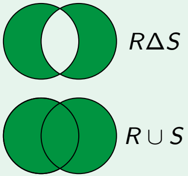
- skills = {Java, C, Python}

Comparison

- Symmetric Set Difference:

$$
\begin{aligned}
R \Delta S &= (R - S) \cup (S - R) \\
&= (R \cup S) - (R \cap S)
\end{aligned}
$$

- Jaccard Distance: $d(R, S) = \frac{|R \Delta S|}{|R \cup S|}$



Composed Data Types: Sets

Bitvector Representation

- Given an ordered base set $B = (b_1, \ldots, b_n)$, for any set S create a binary vector $r \in \{0,1\}^n$ with $r_i = 1 \iff b_i \in S$.
- Hamming distance: Sum of different entries (equals cardinality of symmetric set difference)

Example

- Base: B = (Math, Physics, Chemistry, Biology, Music, Arts, English)
- S = {Math, Music, English} = (1,0,0,0,1,0,1)
- R = {Math, Physics, Arts, English} = (1,1,0,0,0,1,1)
- Hamming(R, S) = 3

Composed Data Types: Sequences, Vectors

Characteristic

- Given a domain D, a sequence s of length n is a mapping $I_n \to D$ of the index set $I_n = \{1, \ldots, n\}$ into D, and we write $s \in D^n$ for short.
- The sequence s concatenates n values from D, and the order does matter

Examples

Distances based on p-norms:	$d_p(o, q) = \|\|o - q\|\|_p = \sqrt[p]{\sum_{i=1}^{n} \|o_i - q_i\|^p}$
$p = 1$: Manhattan distance (city blocks)	$d_1(o, q) = \sum_{i=1}^{n} \|o_i - q_i\|$
$p = 2$: Euclidean distance (aerial, beeline)	$d_2(o, q) = \sqrt{\sum_{i=1}^{n} (o_i - q_i)^2}$
$p \to \infty$: Maximum distance	$d_\infty(o, q) = \max_{i=1}^{n} \|o_i - q_i\|$
Weighted p-distances	$d_{p,w}(o, q) = \sqrt[p]{\sum_{i=1}^{n} w_i \cdot \|o_i - q_i\|^p}$

Complex Data Types

Components

- Structure: graphs, networks, trees
- Geometry: shapes, contours, routes, trajectories
- Multimedia: images, audio, text, etc.

Similarity models: Approaches

- Direct measures – highly data type dependent
- Feature engineering – explicit vector space embedding with hand-crafted features
- Feature learning – explicit vector space embedding learned by machine learning methods, e.g. neural network
- Kernel trick – implicit vector space embedding

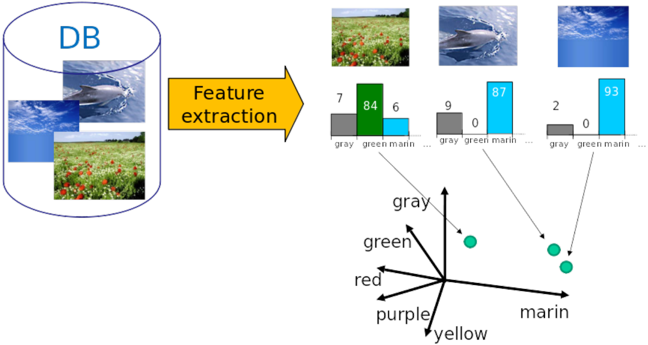
Complex Data Types

Examples for similarity models

	Direct	Feature engineering	Feature learning	Kernel-based
Graphs	Structural Alignment	Degree Histograms	Node embeddings	Label Sequence Kernel
Geometry	Hausdorff Distance	Shape Histograms	Spectral Neural Network	Spatial Pyramid Kernel
Sequences	Edit Distance	Symbol Histograms	Recurrent neural network (RNN)	Cosine Distance

Feature Extraction

- Objects from database DB are mapped to feature vectors



- Feature vector space
  - Points represent objects
  - Distance corresponds to (dis-)similarity

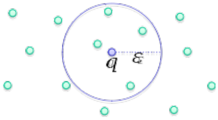
Similarity Queries

- Similarity queries are basic operations in (multimedia) databases
- Given: universe O, database $DB \subseteq O$, distance function d and query object $q \in O$

Range query

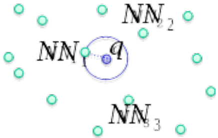
Range query for range parameter $\varepsilon \in \mathbb{R}_0^+$:

$$range(DB, q, d, \varepsilon) = \{o \in DB \mid d(o, q) \leq \varepsilon\}$$



Nearest neighbor query

$$NN(DB, q, d) = \{o \in DB \mid \forall o' \in DB : d(o, q) \leq d(o', q)\}$$



Similarity Queries

k-nearest neighbor query

k-nearest neighbor query for parameter $k \in \mathbb{N}$:

$$NN(DB, q, d, k) \subset DB \text{ with } |NN(DB, q, d, k)| = k \text{ and}$$

$$\forall o \in NN(DB, q, d, k), o' \in DB - NN(DB, q, d, k) : d(o, q) \leq d(o', q)$$

Ranking query

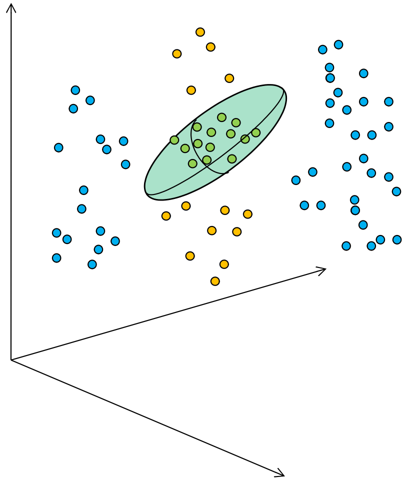
Ranking query (partial sorting query): "get next" functionality for picking database objects in an increasing order w.r.t. their distance to q:

$$\forall i \leq j : d(q, rank_{DB,q,d}(i)) \leq d(q, rank_{DB,q,d}(j))$$

Similarity Search

- Example: Range query $range(DB, q, d, \varepsilon) = \{o \in DB \mid d(o, q) \leq \varepsilon\}$
- Naive search by sequential scan
  - Fetch database objects from secondary storage (e.g. disk): $O(n) time$
  - Check distances individually: $O(n) time$
- Fast search by applying database techniques
  - Filter-refine architecture
    - Filter: Boil database DB down to (small) candidate set $C \subseteq DB$
    - Refine: Apply exact distance calculation to candidates from C only
  - Indexing structures
    - Avoid sequential scans by (hierarchical or other) indexing techniques
    - Data access in time $O(n)$, $O(\log n)$, or even $O(1)$

Filter-Refine Architecture



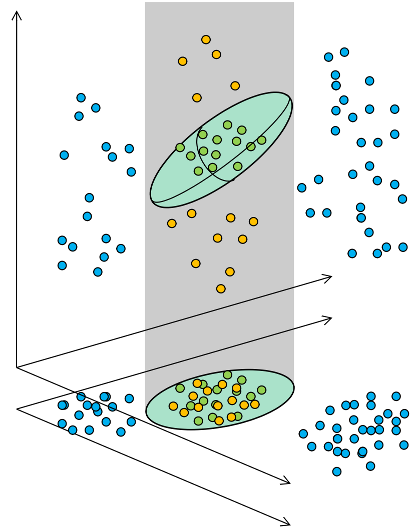
- Principle of multi-step search:
  1. Fast filter step produces candidate set $C \subset DB$ (by approximate distance function d')
  2. Exact distance function d is calculated on candidate set C only.
- Example: Dimensionality reduction[a]
- ICES criteria for filter quality[b]
  - I ndexable – Index enabled
  - C omplete – No false dismissals
  - E fficient – Fast individual calculation
  - S elective – Small candidate set

---

[a] GEMINI: Faloutsos 1996; KNOP: Seidl & Kriegel 1998

[b] Assent, Wenning, Seidl: ICDE 2006

Filter-Refine Architecture



- Principle of multi-step search:
  1. Fast filter step produces candidate set $C \subset DB$ (by approximate distance function d')
  2. Exact distance function d is calculated on candidate set C only.
- Example: Dimensionality reduction[a]
- ICES criteria for filter quality[b]
  - I ndexable – Index enabled
  - C omplete – No false dismissals
  - E fficient – Fast individual calculation
  - S elective – Small candidate set

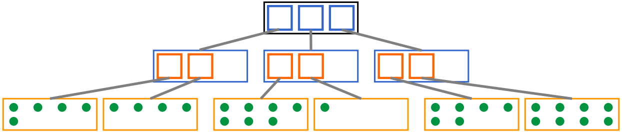
---

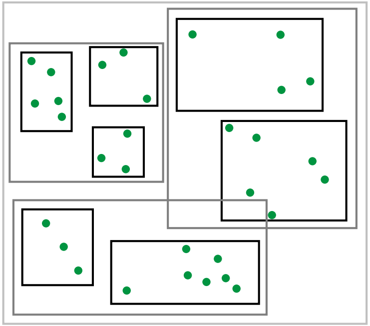
[a] GEMINI: Faloutsos 1996; KNOP: Seidl & Kriegel 1998

[b] Assent, Wenning, Seidl: ICDE 2006

Indexing

- Organize data in a way that allows for fast access to relevant objects, e.g. by heavy pruning.





- R-Tree as an example for spatial index structure:
  - Hierarchy of minimum bounding rectangles
  - Disregard subtrees which are not relevant for the current query region

Indexing

- Example: Phone book
- Indexed using alphabetical order of participants
- Instead of sequential search:
  - Estimate region of query object (interlocutor)
  - Check for correct branch
  - Use next identifier of query object
  - Repeat until query is finished

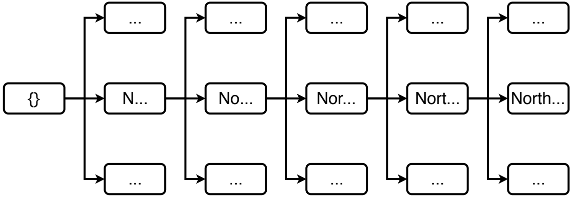


Image source: hierher/flickr, Licence: CC BY 2.0

Agenda

Data Reduction

Why data reduction?

- Better perception of patterns
  - Raw (tabular) data is hard to understand
  - Visualization is limited to (hundreds of) thousands of objects
  - Reduction of data may help to identify patterns
- Computational complexity
  - Big data sets cause prohibitively long runtime for data mining algorithms
  - Reduced data sets are useful the more the algorithms produce (almost) the same analytical results

How to approach data reduction?

- Data aggregation (basic statistics)
- Data generalization (abstraction to higher levels)

Data Reduction Strategies: Three Directions

ID	A1	A2	A3
1	54	56	75
2	87	12	65
3	34	63	76
4	86	23	4

Numerosity Reduction
Reduce number of objects

Dimensionality Reduction
Reduce number of attributes

Quantization, Discretization
Reduce number of values per domain

ID	A1	A3
1	L	75
3	XS	76
4	XL	4

Numerosity reduction

Reduce number of objects

- Sampling (loss of data)
- Aggregation (model parameters, e.g., center and spread)

Data Reduction Strategies (cont'd)

Dimensionality reduction

Reduce number of attributes

- Linear methods: feature sub-selection, Principal Components Analysis, Random projections, Fourier transform, Wavelet transform, etc
- Non-linear methods: Multidimensional scaling (force model), Neural embedding

Quantization, discretization

Reduce number of values per domain

- Binning (various types of histograms)
- Generalization along hierarchies (OLAP, attribute-oriented induction)

Data Aggregation

- Aggregation is numerosity reduction (= less tuples)
- Generalization yields duplicates $\rightarrow$ add counting attribute and merge duplicate tuples

Name	Age	Major
Ann	27	CS
Bob	26	CS
Eve	19	CS

Generalization

Name	Age	Major
(any)	Twen	CS
(any)	Twen	CS
(any)	Teen	CS

Aggregation

Age	Major	Count
Twen	CS	2
Teen	CS	1

Basic Aggregates

- Central tendency: Where is the data located? Where is it centered?
  - Examples: mean, median, mode, etc. (see below)
- Variation, spread: How much do the data deviate from the center?
  - Examples: variance / standard deviation, min-max-range, ...

Examples

- Age of students is around 20
- Shoe size is centered around 40
- Recent dates are around 2020
- Average income is in the thousands

Distributive Aggregate Measures

Distributive Measures

The value of a *distributive measure* d on D can be calculated by combining the results of distributed calculations on partitions $D_i \subset D, D = D_1 \cup D_2 \cup \ldots D_n$

Examples

- $count(D_1 \cup D_2) = count(D_1) + count(D_2)$
- $sum(D_1 \cup D_2) = sum(D_1) + sum(D_2)$
- $min(D_1 \cup D_2) = min(min(D_1), min(D_2))$
- $max(D_1 \cup D_2) = max(max(D_1), max(D_2))$

Algebraic Aggregate Measures

Algebraic Measures

An *algebraic measure* on D can be computed by an algebraic function with M arguments (M being a bounded integer), each of which is obtained by applying a distributive aggregate function to the partitions $D_i \subset D, D = D_1 \cup D_2 \cup \ldots D_n$

Examples

- $avg(D_1 \cup D_2) = \frac{sum(D_1 \cup D_2)}{count(D_1 \cup D_2)} = \frac{sum(D_1)+sum(D_2)}{count(D_1)+count(D_2)}$
- Note: *avg* is note distributive, $avg(D_1 \cup D_2) \neq avg(avg(D_1), avg(D_2))$
- $standard\_deviation(D_1 \cup D_2)$

Holistic Aggregate Measures

Holistic Measures

There is no constant bound on the storage size that is needed to calculate and represent sub-aggregates.

Examples

- *median*: value in the middle of a sorted series of values (=50% quantile)

$$median(D_1 \cup D_2) \neq simple\_function(median(D_1), median(D_2))$$

- *mode*: value that appears most often in a set of values
- *rank*: k-smallest / k-largest value (cf. quantiles, percentiles)

Measuring the Central Tendency

Mean – (weighted) arithmetic mean

Well-known measure for central tendency ("average").

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

Mid-range

Average of the largest and the smallest values in a data set:

$$(max + min)/2$$

- Both are algebraic measures
- Applicable to numerical data only (sum, scalar multiplication)

What about categorical data?

Measuring the Central Tendency (cont'd)

Median

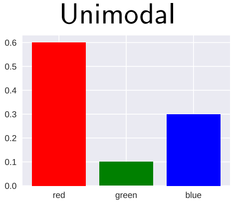
- Middle value of sorted values (if their count is odd)
- For even number of values: average of the middle two values (numeric case), or one of the two middle values (non-numeric case)
- Applicable to ordinal data only, as a (total) order is required
- Median is a holistic measure

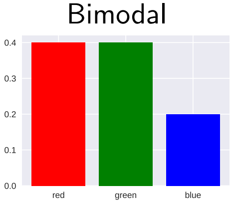
Examples

- never, never, never, rarely, <u>rarely</u>, often, usually, usually, always
- tiny, small, big, big, <u>big</u>, <u>big</u>, big, big, huge, huge
- tiny, tiny, small, <u>medium</u>, <u>big</u>, big, large, huge

What if there is no ordering?

Measuring the Central Tendency





Mode

- Value that occurs most frequently in the data
- Example: *blue*, red, *blue*, yellow, green, *blue*, red
- Unimodal, bimodal, trimodal, ...: There are 1, 2, 3, ... modes in the data (multi-modal in general), cf. mixture models
- There is no mode if each data value occurs only once
- Well suited for categorical (i.e., non-numerical) data

Measuring the Dispersion of Data

Variance

- The variance measures the spread around the mean of numerical data:

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

- The single pass calculation (sum of squares and square of sum in parallel) is faster than the two-pass method but numerically less robust in case of big numbers.
- Variance is zero if and only if all the values are equal
- Standard deviation is equal to the square root of the variance
- Both the standard deviation and the variance are algebraic measures

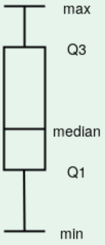
Boxplot Analysis

Boxplots comprise a five-number summary of a dataset

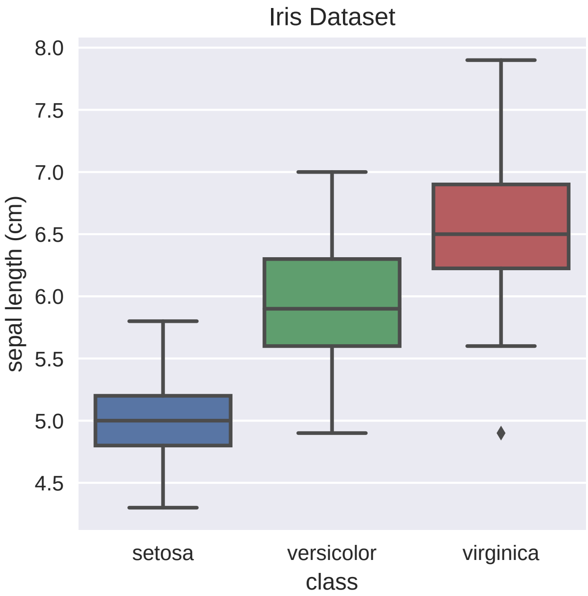
- Minimum, first quartile, median, third quartile, maximum
- These are the 0%, 25%, 50%, 75%, 100% quantiles of the data
- Also called "25-percentile", etc.

Boxplot illustration

- The box ranges from the first to the third quartile
- Height: inter-quartile range (IQR) = Q3 - Q1
- The median is marked by a line within the box
- Whiskers at minimum and maximum value
- Outliers: usually values more than $1.5 \cdot IQR$ below Q1 or above Q3

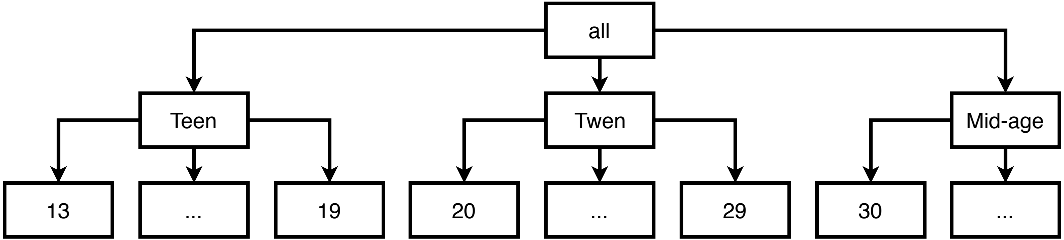


Boxplot Example



Data Generalization

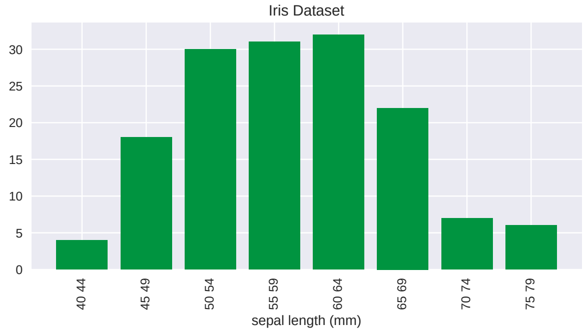
- Quantization may use generalization
  - E.g., group age (7 bits) to age_range (4 bits)
- Dimensionality reduction is a border case of quantization
  - Dropping age reduces 7 bits to zero bits
  - Corresponds to generalization of age to "all" = "any age" = no information



Data Generalization

- How to group the values of an attribute into partitions?
- All data
  - Overall mean, overall variance: too coarse (overgeneralized)
- Different techniques to form groups for aggregation
  - Binning – histograms, based on value ranges
  - Generalization – abstraction based on generalization hierarchies
  - Clustering (see later) – based on object similarity

Binning Techniques: Histograms



- Histograms use binning to approximate data distributions
- Divide data into bins and store a representative (sum, average, median) for each bin

Equi-width Histograms

- Divide 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 $(B - A)/N$

Positive

- Most straightforward

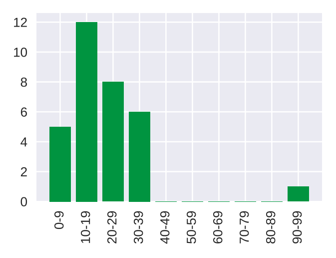
Negative

- Outliers may dominate presentation
- Skewed data is not handled well

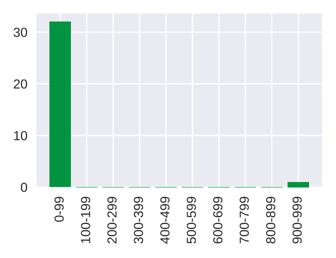
Equi-width Histograms

Example

- Sorted data, 10 bins: 5, 7, 8, 8, 9, 11, 13, 13, 14, 14, 14, 15, 17, 17, 17, 18, 19, 23, 24, 25, 26, 26, 26, 27, 28, 32, 34, 36, 37, 38, 39, 97



- Insert 999



Equi-height Histograms

Divide the range into N intervals, each containing approx. the same number of samples (*quantile-based approach*)

Positive

- Good data scaling

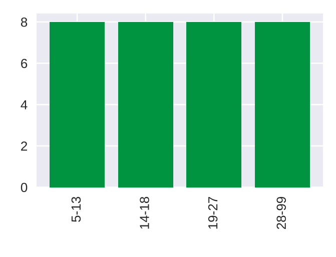
Negative

- If any value occurs often, the equal frequency criterion might not be met (intervals have to be disjoint!)

Equi-height Histograms

Example

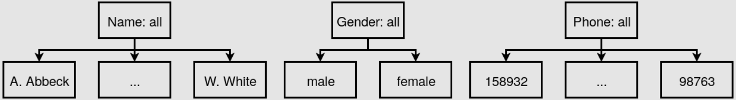
- Same data, 4 bins: 5, 7, 8, 8, 9, 11, 13, 13, 14, 14, 14, 15, 17, 17, 17, 18, 19, 23, 24, 25, 26, 26, 26, 27, 28, 32, 34, 36, 37, 38, 39, 97



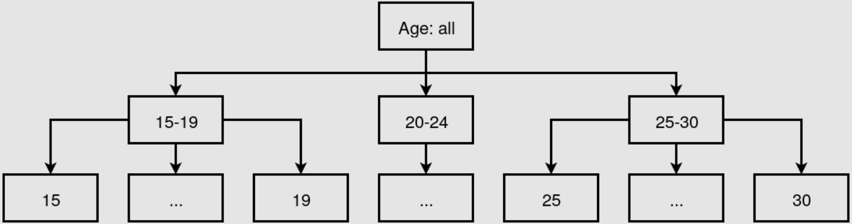
- Median = 50%-quantile
  - More robust against outliers (cf. value 999 from above)
  - Four bin example is strongly related to boxplot

Concept Hierarchies: Examples

No (real) hierarchies

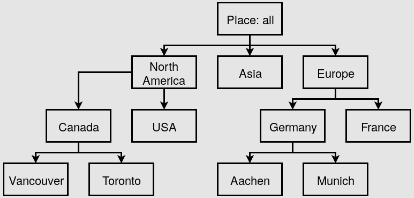


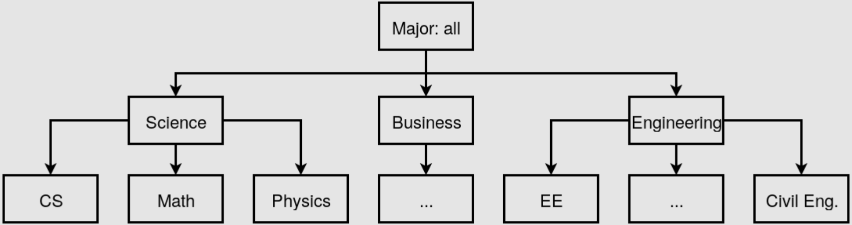
Hierarchies by subgrouping



Concept Hierarchies: Examples

Hierarchies from schema information

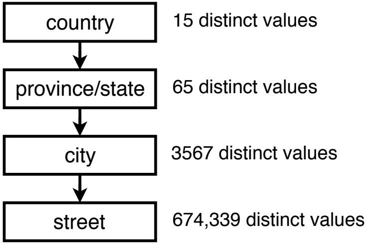




Concept Hierarchy for Categorical Data

- Concept hierarchies can be specified by experts or just by users

- Heuristically generate a hierarchy for a set of (related) attributes
  - based on the number of distinct values per attribute in the attribute set
  - The attribute with the most distinct values is placed at the lowest level of the hierarchy

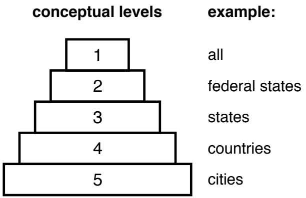


- Fails for counter examples: 20 distinct years, 12 months, 7 days_of_week, but not "year $<$ month $<$ days_of_week" with the latter on top

Summarization-based Aggregation

Data Generalization

A process which abstracts a large set of task-relevant data in a database from low conceptual levels to higher ones.



- Approaches:
  - Data-cube approach (OLAP / Roll-up) – manual
  - Attribute-oriented induction (AOI) – automated

Basic OLAP (Online Analytical Processing) Operations

Roll up

Summarize data by climbing up hierarchy or by dimension reduction.

Drill down

Reverse of roll-up. From higher level summary to lower level summary or detailed data, or introducing new dimensions.

Slice and dice

Selection on one (slice) or more (dice) dimensions.

Pivot (rotate)

Reorient the cube, visualization, 3D to series of 2D planes.

Example: Roll up / Drill down

Query

```
SELECT country, quarter, some_agg_fnc(...)
FROM business
GROUP BY country, quarter
```

Roll-Up

```
SELECT continent, quarter, some_agg_fnc(...)
FROM business
GROUP BY continent, quarter
```

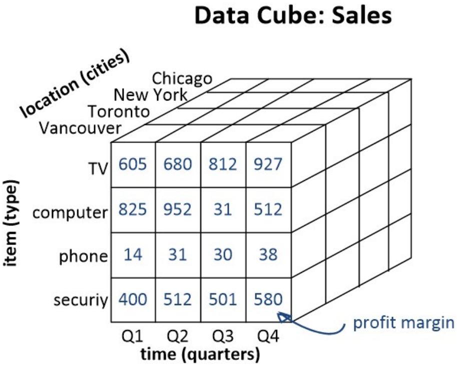
```
SELECT country, some_agg_fnc(...)
FROM business
GROUP BY country
```

Drill-Down

```
SELECT city, quarter, some_agg_fnc(...)
FROM business
GROUP BY city, quarter
```

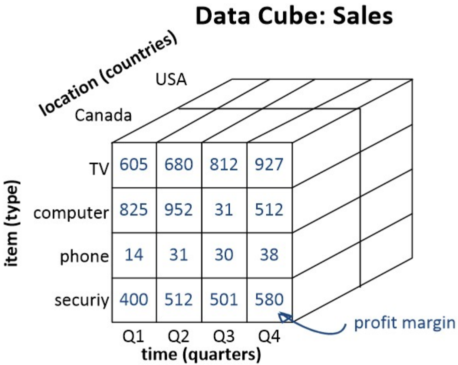
```
SELECT country, quarter, product, some_agg_fnc(...)
FROM business
GROUP BY country, quarter, product
```

Example: Roll up and Drill-Down in a Data Cube



Roll Up $\Longrightarrow$

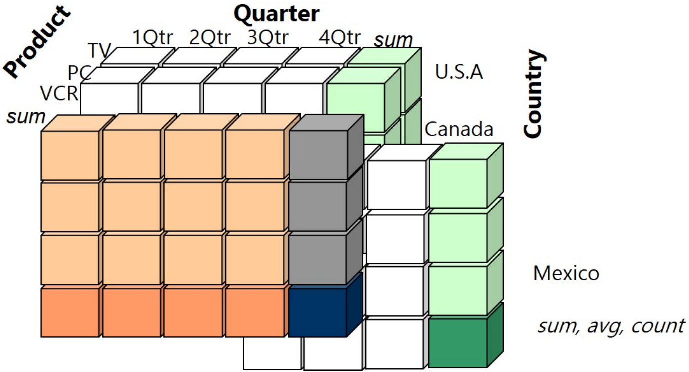
Drill-Down $\Longleftarrow$



Example: Slice Operation

```
SELECT income
FROM   time t, product p, country c
WHERE  p.name = 'VCR'
```

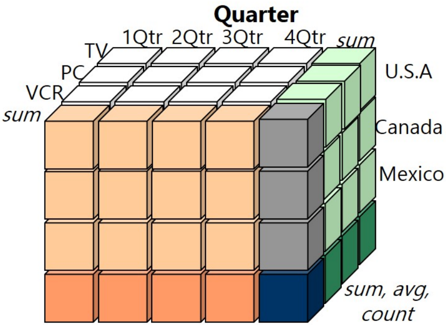
VCR dimension is chosen



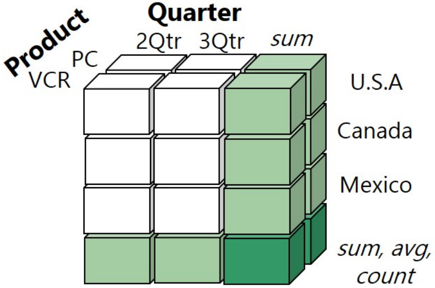
Example: Dice Operation

```
SELECT income
FROM   time t, product p, country c
WHERE  p.name = 'VCR' OR p.name = 'PC' AND t.quarter BETWEEN 2 AND 3
```

sub-data cube over PC, VCR and quarters 2 and 3 is extracted



Dice
$\Longrightarrow$



Example: Pivot (rotate)

year	17			18			19		
product	TV	PC	VCR	TV	PC	VCR	TV	PC	VCR
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

↓ Pivot (rotate) ↓

product	TV			PC			VCR		
year	17	18	19	17	18	19	17	18	19
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

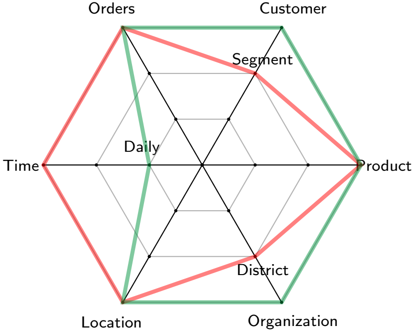
Further OLAP Operations

Other operations

- *Drill across:* involving (across) more than one fact table
- *Drill through:* through the bottom level of the cube to its back-end relational tables (using SQL)

Specifying Generalizations by Star-Nets

- Each circle is called a *footprint*
- Footprints represent the granularities available for OLAP operations



Discussion of OLAP-based Generalization

- Strength
  - Efficient implementation of data generalization
  - Computation of various kinds of measures, e.g., count, sum, average, max
  - Generalization (and specialization) can be performed on a data cube by roll-up (and drill-down)
- Limitations
  - Handles only dimensions of simple non-numeric data and measures of simple aggregated numeric values
  - Lack of intelligent analysis, does not suggest which dimensions should be used and what levels the generalization should aim at

Attribute-Oriented Induction (AOI)

- More automated approach than OLAP
- Apply aggregation by merging identical, generalized tuples and accumulating their respective counts.
- Builds on *data focusing:* task-relevant data, including dimensions, resulting in the *initial relation*
- Generates a *generalization plan:* decide for either *attribute removal* or *attribute generalization*

Attribute-Oriented Induction (AOI)

Three choices for each attribute: keep it, remove it, or generalize it

Attribute Removal

Remove attribute A in the following cases:

- There is a large set of distinct values for A but there is no generalization operator (concept hierarchy) on A, or
- A's higher level concepts are expressed in terms of other attributes (e.g. *street* is covered by *city*, *state*, *country*).

Attribute Generalization

If there is a large set of distinct values for A, and there exists a set of generalization operators (i.e., a concept hierarchy) on A, then select an operator and generalize A.

Attribute Oriented Induction: Example

Name	Gender	Major	Birth place	Birth data	Residence	Phone	GPA
Jim Woodman	M	CS	Vancouver, BC, Canada	8-12-81	3511 Main St., Richmond	687-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-7-80	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-8-75	125 Austin Ave., Burnaby	420-5232	3.83
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

- Name: large number of distinct values, no hierarchy – removed
- Gender: only two distinct values – retained
- Major: many values, hierarchy exists – generalized to Sci., Eng., Biz.
- Birth_place: many values, hierarchy – generalized, e.g., to country
- Birth_date: many values – generalized to age (or age_range)
- Residence: many streets and numbers – generalized to city
- Phone number: many values, no hierarchy – removed
- Grade_point_avg (GPA): hierarchy exists – generalized to good, . . .
- Count: additional attribute to aggregate base tuples

Attribute Oriented Induction: Example

- Initial Relation:

Name	Gender	Major	Birth place	Birth data	Residence	Phone	GPA
Jim Woodman	M	CS	Vancouver, BC, Canada	8-12-81	3511 Main St., Richmond	687-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-7-80	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-8-75	125 Austin Ave., Burnaby	420-5232	3.83
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

- Prime Generalized Relation:

Gender	Major	Birth region	Age Range	Residence	GPA	Count
M	Science	Canada	20-25	Richmond	Very good	16
F	Science	Foreign	25-30	Burnaby	Excellent	22
⋮	⋮	⋮	⋮	⋮	⋮	⋮

- Crosstab for generalized relation:

	Canada	Foreign	Total
M	16	14	30
F	10	22	32
Total	26	36	62

Attribute Generalization Control

- Two common approaches
  - *Attribute-threshold control:* default or user-specified, typically 2-8 values
  - *Generalized relation threshold control:* control the size of the final relation/rule, e.g., 10-30 tuples
- Tradeoff: how many distinct values for an attribute?
  - *Overgeneralization:* values are too high-level
  - *Undergeneralization:* level not sufficiently high
  - Both yield tuples of poor usefulness

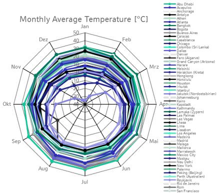
Next Attribute Selection Strategies for Generalization

- Aiming at *minimal degree of generalization*
  - Choose attribute that reduces the number of tuples the most
  - Useful heuristic: choose attribute with highest number of distinct values.
- Aiming at *similar degree of generalization* for all attributes
  - Choose the attribute currently having the least degree of generalization
- *User-controlled*
  - Domain experts may specify appropriate priorities for the selection of attributes

Agenda

Data Visualization

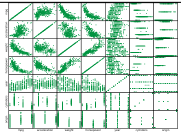
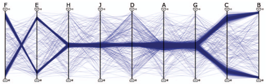
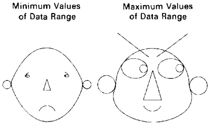
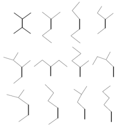
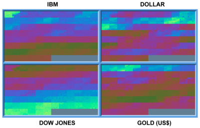
- Patterns in large data sets are hardly perceived from tabular numerical representations
- Data visualization transforms data in visually perceivable representations (”a picture is worth a thousand words”)
- Combine capabilities:
  - Computers are good in number crunching (and data visualization by means of computer graphics)
  - Humans are good in visual pattern recognition



Monthly average temperature [°C]

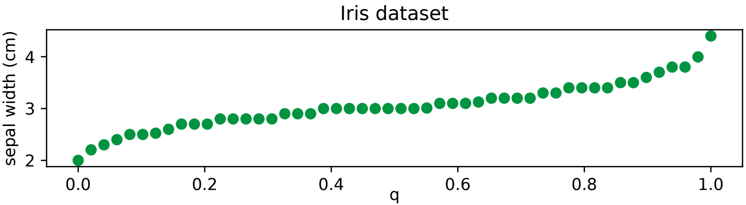
Städte Ø	**Jan**	**Feb**	**Mrz**	**Apr**	**Mai**	**Jun**	**Jul**	**Aug**	**Sep**	**Okt**	**Nov**	**Dez**
Abu Dhabi	25	27	31	36	40	41	42	43	42	37	31	27
Acapulco	32	31	32	32	33	33	33	33	33	33	32	32
Anchorage	-4	-2	0	6	13	17	18	17	13	5	-3	-5
Antalya	15	16	19	22	27	32	35	36	32	27	21	17
Athen	13	14	17	20	26	30	34	34	29	24	18	14
Atlanta	11	13	18	23	26	30	31	31	28	23	17	12
Bangkok	32	33	35	36	35	34	33	33	33	32	32	32
Bogota	20	19	19	19	19	18	18	18	19	19	19	20
Buenos Aires	30	28	26	23	19	16	15	17	19	21	26	29
Caracas	30	28	30	30	31	32	32	32	33	32	31	30
Casablanca	18	18	20	21	22	25	26	27	26	24	21	19
Chicago	0	1	9	16	21	26	29	28	24	17	9	2
Colombo (Sri Lanka)	31	31	32	32	32	31	31	31	31	31	31	31
Dallas	13	16	21	25	29	33	36	36	32	26	19	14
Denver	7	8	14	14	21	28	32	30	25	18	12	6
Faro (Algarve)	16	16	19	21	23	27	29	29	26	23	19	17
Grand Canyon (Arizona)	6	8	13	15	21	27	29	27	25	18	12	6
Harare	27	26	27	26	24	21	22	24	28	29	28	27
Helsinki	-3	-3	2	9	15	20	23	21	17	9	3	0
Heraklion (Kreta)	15	16	18	20	24	27	30	30	27	24	20	17
Hongkong	19	20	23	26	30	32	33	33	32	30	25	21
Honolulu	26	26	27	27	28	30	30	31	30	30	28	27
Houston	16	19	23	27	30	33	34	35	32	28	21	17
Irkutsk	-14	-9	1	9	16	23	24	21	16	7	-4	-13
Istanbul	9	9	13	17	23	27	30	30	26	20	15	11
Jakutsk (Nordostsibirien)	-35	-28	-10	3	14	23	26	21	11	-3	-25	-34
Johannesburg	25	25	24	22	20	17	17	20	24	25	25	25
Kairo	19	20	24	27	32	35	35	35	34	30	25	20
Kapstadt	27	27	26	24	21	18	18	18	19	22	24	26
Kathmandu	18	21	25	28	28	29	28	28	28	26	23	20
Larnaka (Zypern)	17	18	20	23	26	31	33	34	31	28	23	19
Las Palmas	21	20	22	23	24	25	27	28	28	27	24	22
Las Vegas	15	16	23	26	31	38	40	39	35	27	20	14
Lhasa	9	10	13	17	21	24	23	22	21	17	13	10
Lima	26	26	27	24	21	20	19	18	19	20	22	24
Lissabon	14	15	18	20	23	27	28	29	27	22	17	15

Data Visualization Techniques

Type	Idea	Examples
Geometric	Visualization of geometric transformations and projections of the data	 Scatterplots  Parallel Coordinates
Icon-Based	Visualization of data as icons	 Chernoff Faces  Stick Figures
Pixel-oriented	Visualize each attribute value of each data object by one coloured pixel	 Recursive Patterns
Other		Hierarchical Techniques, Graph-based Techniques, Hybrid-Techniques, . . .

Slide credit: Keim, Visual Techniques for Exploring Databases, Tutorial Slides, KDD 1997.

Quantile Plot



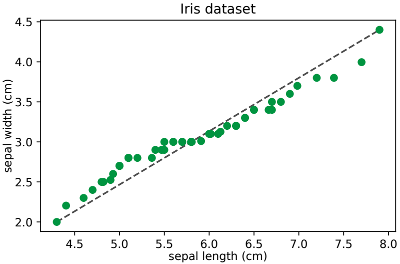
Characteristic

The p-quantile x_p is the value for which the fraction p of all data is less than or equal to x_p.

Benefit

Displays all of the data (allowing the user to assess both the overall behavior and unusual occurrences)

Quantile-Quantile (Q-Q) Plot



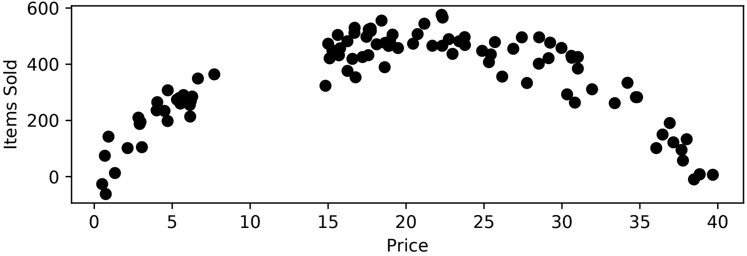
Characteristic

Graphs the quantiles of one univariate distribution against the corresponding quantiles of another.

Benefit

Allows the user to compare to distributions against each other.

Scatter Plot



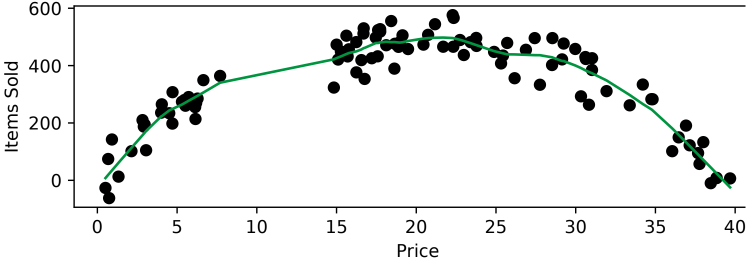
Characteristic

Each pair of values is treated as a pair of coordinates and plotted as points in the plane.

Benefit

Provides a first look at bivariate data to see clusters of points, outliers, etc.

Loess Curve



Characteristic

Loess curve is fitted by setting two parameters: a smoothing parameter, and the degree of the polynomials that are fitted by the regression.

Benefit

Adds a smooth curve to a scatter plot in order to provide better perception of the pattern of dependence.

Scatterplot Matrix

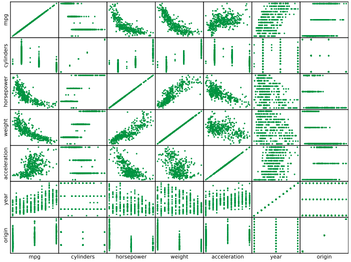
Characteristic

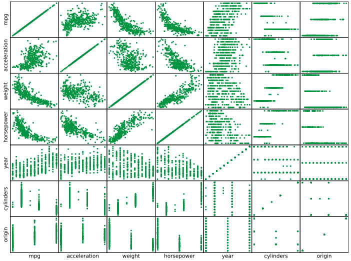
Matrix of scatterplots for pairs of dimensions

Ordering

Ordering of dimensions is important:

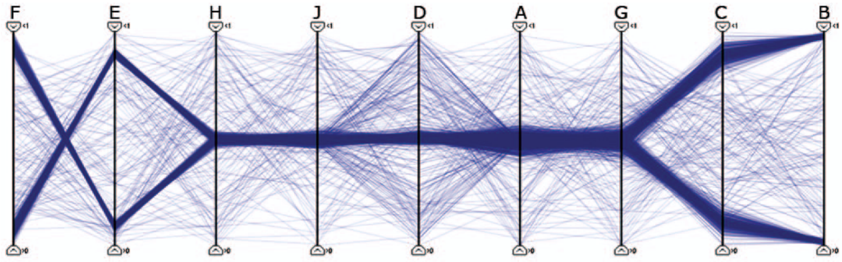
- Reordering improves understanding of structures and reduces clutter
- Interestingness of orderings can be evaluated with quality metrics (e.g. Peng et al.)





Clutter Reduction in Multi-Dimensional Data Visualizazion Using Dimension Reordering, IEEE Symp. on Inf. Vis., 2004.

Parallel Coordinates



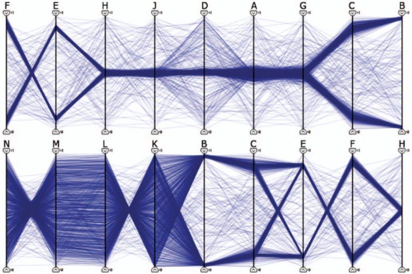
Characteristics

- d-dimensional data space is visualised by d parallel axes
- Each axis is scaled to min-max range
- Object = polygonal line intersecting axis at value in this dimension

Parallel Coordinates

Ordering

- Again, the ordering and orientation of the dimensions is important to reveal correlations
- The farer the attributes the harder their correlations are perceived; coloring helps



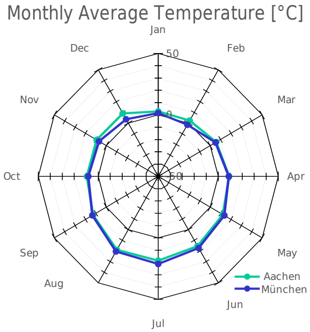
- Visualize clusters
- Visualize correlations between dimensions

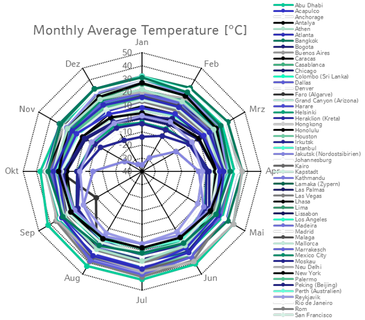
Bertini et al., Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization, Trans. on Vis. and Comp. Graph., 2011.

Spiderweb Model

Characteristics

- Illustrate any single object by a polygonal line
- Contract origins of all axes to a global origin point
- Works well for few objects only





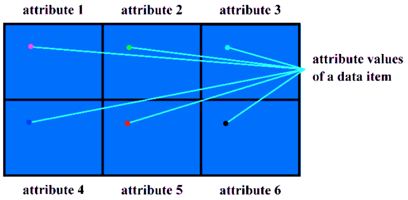
Pixel-Oriented Techniques

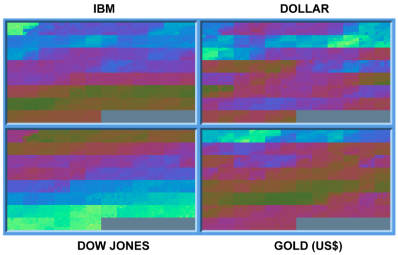
Characteristics

- Each data value is mapped onto a colored pixel
- Each dimension is shown in a separate window

How to arrange the pixel ordering?

One strategy: Recursive Patterns iterated line and column-based arrangements





Figures from Keim, Visual Techniques for Exploring Databases, Tutorial Slides, KDD 1997.

Chernoff Faces

Characteristics

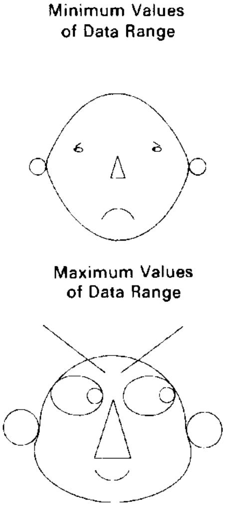
Map d-dimensional space to facial expression, e.g. length of nose = dim 6; curvature of mouth = dim 8

Advantage

Humans can evaluate similarity between faces much more intuitively than between high-dimensional vectors

Disadvantages

- Without dimensionality reduction only applicable to data spaces with up to 18 dimensions
- Which dimension represents what part?



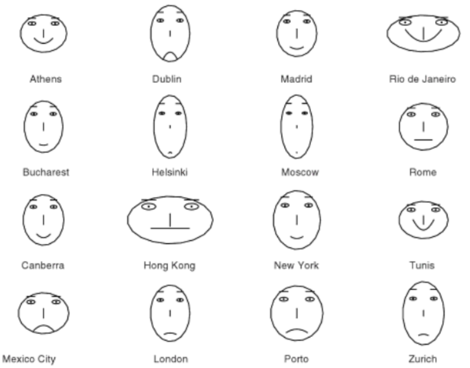
Figures taken from Mazza, Introduction to Information Visualization, Springer, 2009.

Chernoff Faces

Example: Weather Data

City	Precip. average	Temp. average	Temp. max average	Temp. min average	Record max	Record min
Athens	37	17	21	13	42	−3
Bucharest	58	11	16	5	49	−23
Canberra	62	12	19	6	42	−10
Dublin	74	10	12	6	28	−7
Helsinki	63	5	8	1	31	−36
Hong Kong	218	23	25	21	37	2
London	75	10	13	5	35	−13
Madrid	45	13	20	7	40	−10
Mexico City	63	17	23	11	32	−3
Moscow	59	4	8	1	35	−42
New York	118	12	17	8	40	−18
Porto	126	14	18	10	34	−2
Rio de Janeiro	109	25	30	20	43	7
Rome	80	15	20	11	37	−7
Tunis	44	18	23	13	46	−1
Zurich	107	9	12	6	35	−20

Table 4.1 Annual climatic values in Celsius of some world cities. Values from http://www.weatherbase.com.



Figures from Riccardo Mazza, Introduction to Information Visualization, Springer, 2009.

Chernoff Faces

Example: Finance Data

FIGURE 3
Facial Representation of Financial Performance (1 to 5 Years Prior to Failure)

FEDERAL					
Date	**Year to Failure**				
Dimensions	**5**	**4**	**3**	**2**	**1**
1. Return on Assets	0.10	0.11	0.06	0.03	−0.16
2. Debt Service	3.66	3.79	1.55	0.78	−14.11
3. Cash Flows	1.53	1.48	1.39	1.35	0.94
4. Capitalization	0.22	0.20	0.18	0.16	−0.02
5. Current Ratio	71.40	89.10	97.85	96.80	58.21
6. Cash Turnover	24.03	25.92	25.62	27.40	71.26
7. Receivables Turnover	5.25	4.46	4.26	4.36	9.56
8. Inventory Turnover	5.38	4.77	4.57	4.44	5.34
9. Sales per Dollar Working Capital	6.74	6.33	7.02	7.61	−45.77
10. Retained Earning/ Total Assets	0.32	0.30	0.01	−0.01	−0.26
11. Total Assets	0.94	.76	0.39	0.45	0.43

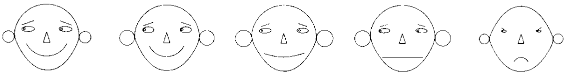


Figure from Huff et al., Facial Representation of Multivariate Data, Journal of Marketing, Vol. 45, 1981, pp. 53-59.

Agenda

Data Privacy

Situation

- Huge volume of data is collected
- From a variety of devices and platforms (e.g. Smartphones, Wearables, Social Networks, Medical systems)
- Capturing human behaviors, locations, routines, activities and affiliations
- Providing an opportunity to perform data analytics

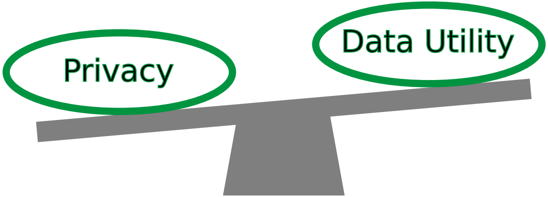
Data Abuse is inevitable

- It compromises individual's privacy
- Or breaches the security of an institution

Data Privacy

- These privacy concerns need to be mitigated
- They have prompted huge research interest to *protect data*
- But,

$$\text{Strong Privacy Protection} \implies \text{Poor Data Utility}$$
$$\text{Good Data Utility} \implies \text{Weak Privacy Protection}$$



Challenge

Find a good trade-off between Data Utility and Privacy

Data Privacy

Objectives of Privacy Preserving Data Mining

- Ensure data privacy
- Maintain a good trade-off between data utility and privacy

Paradigms

- k-Anonymity
- l-Diversity
- Differential Privacy

Linkage Attack

Method

Different public records can be linked to it to breach privacy

Hospital Records

Private	*Public*			
Name	Sex	Age	Zip	Disease
Alice	F	29	52062	Breast Cancer
Janes	F	27	52064	Breast Cancer
Jones	M	21	52066	Lung Cancer
Frank	M	35	52072	Heart Disease
Ben	M	33	52078	Fever
Betty	F	37	52080	Nose Pains

Public Records from Sport Club

Public				
Name	Sex	Age	Zip	Sport
Alice	F	29	52062	Tennis
Theo	M	41	52066	Golf
John	M	24	52062	Soccer
Betty	F	37	52080	Tennis
James	M	34	82066	Soccer

k-Anonymity

k-Anonymity

Privacy paradigm for protecting data records before *publication*

Three kinds of attributes:

1. *Key Attributes*: Uniquely identifiable attributes (e.g., Name, Social Security Number, Telephone Number)
2. *Quasi-identifier*: Groups of attributes that can be combined with external data to uniquely re-identify an individual (e.g. (Date of Birth, Zip Code, Gender))
3. *Sensitive Attributes*: An attacker should not be able to combine these with the key attributes. (e.g. Disease, Salary, Habit, Location etc.)

k-Anonymity

Attention

Hiding key attributes alone does not guarantee privacy.

An attacker may be able to break the privacy by combining the quasi-identifiers from the data with those from publicly available information.

Definition: k-Anonymity

Given a set of quasi-identifiers in a database table, the database table is said to be *k-Anonymous*, if the sequence of records in each quasi-identifier exists at least k times.

Ensure privacy by *Suppression* or *Generalization* of quasi-identifiers.

k-Anonymity: Suppression

Suppression

Accomplished by replacing a part or the entire attribute value by placeholder, e.g. "?" (= generalization)

Example

- Suppress Postal Code: 52062 $\mapsto$ 52???
- Suppress Gender: Male $\mapsto$?; Female $\mapsto$?

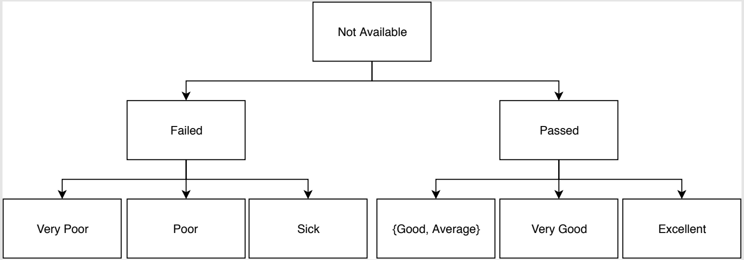
k-Anonymity: Generalization

Generalization

Accomplished by aggregating values from fine levels to coarser resolution using generalisation hierarchy.

Example

Generalize exam grades:



Shortcomings: Background Knowledge Attack

Background Knowledge Attack

Lack of diversity of the sensitive attribute values (homogeneity)

Example

- *Background Knowledge*: Alice is (i) 29 years old and (ii) female
- *Homogeneity*: All 2*-aged females have Breast Cancer.
  $\Longrightarrow$ Alice has BC!

Release

Quasi Identifier			*Sensitive*
Sex	Age	Zip	Disease
F	2?	520??	Breast Cancer
F	2?	520??	Breast Cancer
M	2?	520??	Lung Cancer
M	3?	520??	Heart Disease
M	3?	520??	Fever
F	3?	520??	Nose Pains

This led to the creation of a new privacy model called l-diversity

l-Diversity

Distinct l-Diversity

An quasi-identifier is l-diverse, if there are at least l different values. A dataset is l-diverse, if all QIs are l-diverse.

Example

Not "diverse"

Quasi Identifier	Sensitive
QI 1	Headache
QI 1	Headache
QI 1	Headache
QI 2	Cancer
QI 2	Cancer

2-diverse

Quasi Identifier	Sensitive
QI 1	Headache
QI 1	Cancer
QI 1	Headache
QI 2	Headache
QI 2	Cancer

l-Diversity

Other Variants

- *Entropy l-Diversity*: For each equivalent class, the entropy of the distribution of its sensitive values must be at least l
- *Probabilistic l-Diversity*: The most frequent sensitive value of an equivalent class must be at most $1/l$

Limitations

- Not necessary at times
- Difficult to achieve: For large record size, many equivalent classes will be needed to satisfy l-Diversity
- Does not consider the distribution of sensitive attributes

Background Attack Assumption

- *k*-Anonymity and *l*-Diversity make assumptions about the adversary
- They at times fall short of their goal to prevent data disclosure
- There is another privacy paradigm which does not rely on background knowledge, called *Differential Privacy*

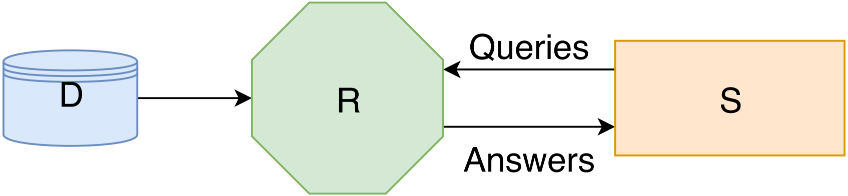
Differential Privacy

Core Idea

Privacy through data perturbation.

- The addition or removal of one record from a database should not reveal any information to an adversary, i.e. your *presence* or *absence* does not reveal or leak any information.
- Use a randomization mechanism to perturb query results of `count`, `sum`, `mean` functions, as well as other statistical query functions.

Differential Privacy



Definition

A randomized mechanism $R(x)$ provides ϵ-*differential privacy* if for any two databases D_1 and D_2 that differ on at most one element, and all outputs $S \subseteq Range(R)$

$$\frac{Pr[R(D_1) \in S]}{Pr[R(D_2) \in S]} \leq \exp(\epsilon)$$

ϵ is a parameter called *privacy budget/level*.

Data Perturbation

Data perturbation is achieved by noise addition.

Some Kinds of Noise

- Laplace noise
- Gaussian noise
- Exponential Mechanism